A Probabilistic Framework for Semi-Autonomous Robots Based on Interaction Primitives with Phase Estimation

Guilherme Maeda, Gerhard Neumann, Marco Ewerton, Rudolf Lioutikov and Jan Peters

Abstract This paper proposes an interaction learning method suited for semi-autonomous robots that work with or assist a human partner. The method aims at generating a collaborative trajectory of the robot as a function of the current action of the human. The trajectory generation is based on action recognition and prediction of the human movement given intermittent observations of his/her positions under unknown speeds of execution; a problem typically found when using motion capture systems in scenarios that lead to occlusion. Of particular interest, the ability to predict the human movement while observing the initial part of his/her trajectory allows for faster robot reactions, and as it will be shown, also eliminates the need of time-alignment of the training data. The method models the coupling between human-robot movement primitives and is scalable in relation to the number of tasks. We evaluated the method using a 7-DoF lightweight robot arm equipped with a 5-finger hand in a multi-task collaborative assembly experiment, also comparing results with our previous method based on time-aligned trajectories.

1 Introduction

Assistive and collaborative robots must have the ability to physically interact with the human, safely and synergistically. However, pre-programming a

Guilherme Maeda[1], Gerhard Neumann[2], Marco Ewerton[1], Rudolf Lioutikov[1] and Jan Peters[1,3]
[1]Intelligent Autonomous Systems Group, TU Darmstadt, Germany
[2]Computational Learning for Autonomous Systems Group, TU Darmstadt, Germany
[3]Max Planck Institute for Intelligent Systems, Tuebingen, Germany
e-mail: {maeda|neumann|ewerton|lioutikov|peters}@ias.tu-darmstadt.de

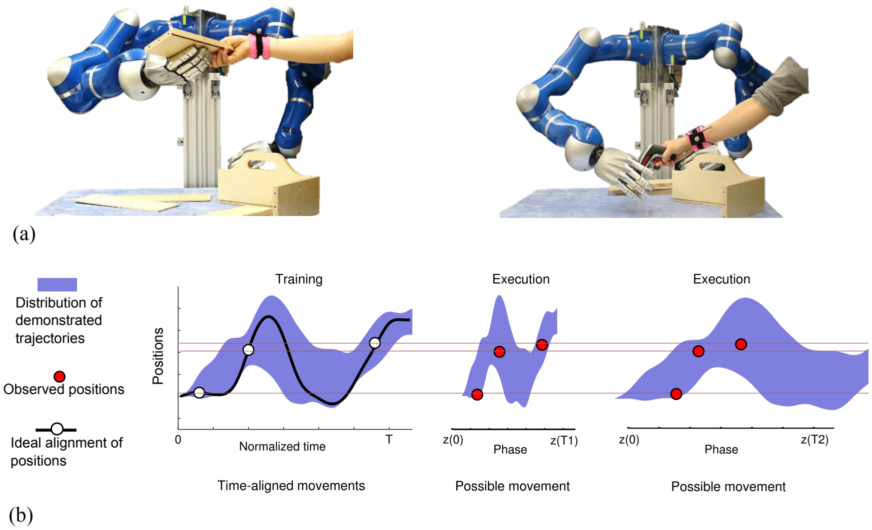


Fig. 1 Collaborative and assistive robots must address both action recognition and movement coordination based on human observations. (a) A robot coworker must recognize the intention of the human before deciding which action to take. (b) Observing the human movement through corrupted (e.g. occluded, sparse, intermittent) position data, poses the problem of identifying the correct phase of the movement.

robot for a large number of tasks is not only tedious, but unrealistic, especially if tasks are added or change constantly. Moreover, conventional programming methods do not address semi-autonomous robots—robots whose actions depend on the actions of a human partner. Nevertheless, once deployed, for example in a domestic or small industrial environment, a semi-autonomous robot must be easy to program, without requiring the need of a dedicated expert. For this reason, we advocate interaction learning as a data-driven approach based on the use of imitation learning [18] for human-robot interaction.

Amongst the several challenges posed by interaction learning, this paper focuses on two intrinsically related problems. First, the problem of estimating the phase of the human movement, that is, the progress or the stage of the execution of his/her trajectory under an intermittent streaming of position data. This is a problem of practical importance since the majority of motion capture methods currently available, such as marker tracking and depth cameras, rely on planned spaces and well positioned cameras; requirements that are incompatible with most already existing collaborative environments of interest (e.g. in a hospital) where occlusions are prone to occur. Second, based on this assessment, we address the problem of recognizing the action and generating the corresponding movement of the robot assistant. As illustrated in Fig. 1(a), by observing the movement of the human, a semi-autonomous robot must decide if it should hand over a plate, or hold a screwdriver. The human, however, may execute movements at different unobserved speeds, and position measurements may be corrupted by occlusions, which causes the problem of temporally aligning sparse position observations with the interaction model. Fig. 1(b) illustrates such a problem where the same sequence of three observed positions may fit two models that are identically spatially, but with different phases. Such ambiguity clearly hinders the adaptation of the robot movement.

The contribution of this paper is a probabilistic framework for interaction learning with movement primitives that allows a robot to react faster by estimating the phase of the human, and to relate the outcome of the estimation to assess different tasks. As the algorithm relies on Probabilistic Movement Primitives [16] for human-robot interaction, the method will also be referred to as Interaction ProMPs. An Interaction ProMP provides a model that correlates the weights that parameterize the trajectories of a human and a robot when executing a task in collaboration. The Interaction ProMP is conditioned on the observations of the human and the robot is controlled based on the posterior distribution over robot trajectories.

This paper consolidates our recent efforts in different aspects of semi-autonomous robots. It leverages on the representation of movements with ProMPs, our developments into the context of human-robot interaction [1, 14], and the ability to address multiple tasks [14, 8]. While our previous interaction models were explicitly time-dependent, here, we introduce a phase-dependent method. Section 2 emphasizes the most relevant works in phase and time representations and briefly addresses related works in other aspects of the framework[1]. Section 3 describes the proposed method with a brief background on ProMPs, followed by Interaction ProMPs, phase estimation, and action recognition. Finally, Section 4 provides experiments and discussions on the application of the method in an assembly scenario.

2 Related Work

Dynamical Movement Primitives [9], or simply DMPs, have been known to address temporal variations with a phase variable. The phase variable is used to govern the spread of a fixed number of basis functions that encode parameters of a forcing function. ProMPs use the concept of phases in the same manner, with the difference that the basis functions are used to encode positions. This difference is fundamental for Interaction Primitives since estimating the forcing function of the human is nontrivial in practice, while positions can be often measured directly [14].

[1] The interested reader is referred to our previous works for additional and detailed literature review in respect to their corresponding contributions.

Recently, a modified form of DMPs where the rate of phase change is related to the speed of movement has been presented [21]. The method uses Reinforcement Learning and Iterative Learning Control to speed up the execution of a robot's movement without violating pre-defined constraints such as carrying a glass full of liquid without spilling it. A similar form of iterative learning was used to learn the time mapping between demonstrated trajectories and a reference trajectory [20]. With their approach, a robot was able to perform a surgical task of knot-tie faster than the human demonstrator.

Dynamic Time Warping (DTW) [17] has been used in robotics applications for temporally aligning trajectories. For example, as part of an algorithm that estimates the optimal, hidden trajectory provided by multiple expert demonstrations [5]. Although DTW has been shown suitable for off-line processing of data, its online application can be hard to achieve in practice due to exhaustive systematic search. A different approach is to explicitly encode the time of demonstrations such as in [3], where the structure of the model intrinsically generate smooth temporal solutions. The measurement or estimation of velocity, for example, by differentiation of a consistent stream of positions, removes the ambiguity of Fig. 1(b) and allows for the realization of online algorithms that cope with very fast dynamics [10, 11]. Such methods, however, rely on a planned environment free from occlusions and fast tracking capabilities; requirements difficult to achieve in environments where semi-autonomous robots are expected to make their biggest impact, such as in small factories, hospitals and home care facilities. A disadvantage of ProMPs in relation to representations based on multiple reference frames such as the Dynamical Systems [3], and forcing functions as in DMPs, is that ProMPs can only operate within the demonstrated, spatially invariant set of demonstrations.

Several methods to learn time-independent models by imitation have been proposed. For example, Hidden Markov Models (HMM) and Gaussian Mixture Regression (GMR) have been used to learn and reproduce demonstrated gestures [4] where each hidden state corresponds to a Gaussian over positions and velocities, locally encoding variation and correlation. In [6], a method to reactively adapt trajectories of a motion planner due to changes in the environment was proposed by measuring the progress of a task with a dynamic phase variable. While this method is suited for cases where the goal is known from a planned trajectory—the phase is estimated from the distance to the goal—a semi-autonomous robot is not provided with such information: the goal must be inferred from the observation of the human movement, which in turn requires an estimate of the phase.

This paper share similar challenges faced in [22] where the robot trajectory had to be adapted according to the observation of the human partner during handovers. In their work, the authors encoded the demonstrations in a tree-structured database as a hierarchy of clusters, which poses the problem of searching matching trajectories given partial observations. The use of a probabilistic approach allow us to address an equivalent search by simply computing the likelihoods of various models given arbitrary segments of observed trajectories.

Several other works have addressed the action recognition problem. Graphical models, in particular, have been widely used. In human-robot interaction, HMMs have been used hierarchically to represent the states and to trigger low-level primitives [13]. HMMs were also applied to predict the positions of a coworker in an assembly line for tool delivery [19] while in [12], Conditional Random Fields were used to predict the possible actions of a human. The prediction of the movement of human coworkers was addressed in [15] with a mixture model. The cited methods address the generation of the corresponding robot movement as an independent step, either by pre-programming suitable actions [12], or by using motion planners [15]. In a similar vein, Petri Nets were combined to control the transitions of a controller for a robot mounted on the shoulder of a human [2]. Interaction ProMPs intrinsically correlate the action of the human with the movement of the robot such that action recognition and movement generation are part of the same process and use the same model.

3 Probabilistic Movement Primitives for Human-Robot Interaction

This section introduces ProMPs for a single degree-of-freedom (DoF) from which the multi-DoF ProMP will follow naturally. In human-robot interaction, the use of ProMPs consists on the adaptation of the multi-DoF case where some of the DoFs are given by a tracked human interacting with a semi-autonomous robot. This section finishes by introducing phase estimation which will also provide means to recognize human actions in multiple-task scenarios.

3.1 Probabilistic Movement Primitives on a Single Degree-of-Freedom

For each time step t a position is represented by $\boldsymbol{y}_t$ and a trajectory of length T as a smooth sequence $\boldsymbol{y}_{0:T}$. A parameterization of $\boldsymbol{y}_{0:T}$ in a lower dimensional weight space can be achieved by linear regression on time-dependent Gaussian basis functions $\boldsymbol{\psi}_t$,

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

$$p(\boldsymbol{y}_{0:T}|\boldsymbol{w}) = \prod_{0}^{T} \mathcal{N}(\boldsymbol{y}_t|\boldsymbol{\psi}_t^T \boldsymbol{w}, \boldsymbol{\Sigma}_y), \tag{2}$$

where $\boldsymbol{\epsilon}_y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_y)$ is zero-mean i.i.d. Gaussian noise and $\boldsymbol{w} \in \mathbb{R}^N$ is a weight vector that encodes the trajectory. The number of Gaussian bases N is often much lower than the number of trajectory time steps. In our case, trajectories have an average time of 3 seconds and are sampled at 50 Hz. The dimensionality is decreased from $3 \times 50 = 150$ samples to a vector of length $N = 20$. The number of basis is a design parameter that must be matched with the desired amount of detail to be preserved during the encoding of the trajectory.

Assume M trajectories are obtained via demonstrations; their parameterization leading to a set of weight vectors $\boldsymbol{W} = \{\boldsymbol{w}_1, \ ... \ \boldsymbol{w}_i, \ ... \ \boldsymbol{w}_M\}$ (the subscript i as in $\boldsymbol{w}_i$ will be used to indicate a particular demonstration when relevant, and will be omitted otherwise). Define $\boldsymbol{\theta}$ as a parameter to govern the distribution of $\boldsymbol{W}$ such that $\boldsymbol{w} \sim p(\boldsymbol{w}; \boldsymbol{\theta})$. From the training data we model $p(\boldsymbol{w}; \boldsymbol{\theta})$ as a Gaussian with mean $\boldsymbol{\mu}_w \in \mathbb{R}^N$ and covariance $\boldsymbol{\Sigma}_w \in \mathbb{R}^{N \times N}$, that is $\boldsymbol{\theta} = \{\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w\}$. This model allow us to sample from the demonstrated distribution over positions with

$$p(\boldsymbol{y}_t; \boldsymbol{\theta}) = \int p(\boldsymbol{y}_t|\boldsymbol{w}) p(\boldsymbol{w}; \boldsymbol{\theta}) d\boldsymbol{w} = \mathcal{N}(\boldsymbol{y}_t|\boldsymbol{\psi}_t^T \boldsymbol{\mu}_w, \boldsymbol{\psi}_t^T \boldsymbol{\Sigma}_w \boldsymbol{\psi}_t + \boldsymbol{\Sigma}_y). \tag{3}$$

The Gaussian assumption is restrictive in two ways. First, the training data must be time-aligned, for example by DTW; second, only one type of interaction pattern—or collaborative task—can be encoded within a single Gaussian (mixture of models were used to address the latter problem in an unsupervised fashion [8]).

3.2 Correlating Human and Robot Movements with Interaction ProMPs

Interaction ProMPs model the correlation of multiple DoFs of multiple agents. Let us define the state vector as a concatenation of the P number of observed DoFs of the human, followed by the Q number of DoFs of the robot

$$\boldsymbol{y}_t = [\ \boldsymbol{y}_{1,t}^H, \cdots \ \boldsymbol{y}_{P,t}^H, \ \boldsymbol{y}_{1,t}^R, \cdots \ \boldsymbol{y}_{Q,t}^R \ ]^T,$$

where the upper scripts $(\cdot)^H$ and $(\cdot)^R$ refer to the human and robot DoFs, respectively. Similar to the single DoF case, all DoF's trajectories are parameterized as weights such that

$$p(\boldsymbol{y}_t|\bar{\boldsymbol{w}}) = \mathcal{N}(\boldsymbol{y}_t|\mathbf{H}_t^T \bar{\boldsymbol{w}}, \boldsymbol{\Sigma}_y), \tag{4}$$

where $\mathbf{H}_t^T = \mathrm{diag}((\boldsymbol{\psi}_t^T)_1, ..., (\boldsymbol{\psi}_t^T)_P, (\boldsymbol{\psi}_t^T)_{P+1}, ..., (\boldsymbol{\psi}_t^T)_{P+Q})$ has $P{+}Q$ diagonal entries. Each collaborative demonstration now provides $P{+}Q$ training trajectories. The weight vector $\bar{\boldsymbol{w}}_i$ of the i-th demonstration is

$$\bar{\boldsymbol{w}}_i = [\ (\boldsymbol{w}_1^H)^T, ..., (\boldsymbol{w}_P^H)^T,\ (\boldsymbol{w}_1^R)^T, ..., (\boldsymbol{w}_Q^R)^T\ ]^T, \tag{5}$$

from which a normal distribution from a set of M demonstrations $\bar{\boldsymbol{W}} = \{\bar{\boldsymbol{w}}_1, ...\bar{\boldsymbol{w}}_M\}$ with $\boldsymbol{\mu}_w \in \mathbb{R}^{(P+Q)N}$ and $\boldsymbol{\Sigma}_w \in \mathbb{R}^{(P+Q)N \times (P+Q)N}$ can be computed.

The fundamental operation for semi-autonomy is to compute a posterior probability distribution of the weights (now encoding both human and robot) $\bar{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{\mu}_w^{new}, \boldsymbol{\Sigma}_w^{new})$ conditioned on a sparse (e.g. due to motion capture occlusion) sequence of observed positions of the human $\boldsymbol{y}^*$ measured within the interval $[t, t']$. This operation can be computed with

$$\begin{aligned} \boldsymbol{\mu}_w^{new} &= \boldsymbol{\mu}_w + \boldsymbol{K}(\boldsymbol{y}_{t,t'}^* - \mathbf{H}_{t,t'}^T \boldsymbol{\mu}_w), \\ \boldsymbol{\Sigma}_w^{new} &= \boldsymbol{\Sigma}_w - \boldsymbol{K}(\mathbf{H}_{t,t'}^T \boldsymbol{\Sigma}_w), \end{aligned} \tag{6}$$

where $\boldsymbol{K} = \boldsymbol{\Sigma}_w \mathbf{H}_{t,t'}^T (\boldsymbol{\Sigma}_y^* + \mathbf{H}_{t,t'}^T \boldsymbol{\Sigma}_w \mathbf{H}_{t,t'})^{-1}$ and $\boldsymbol{\Sigma}_y^*$ is the measurement noise. The upper-script $(\cdot)^{new}$ is used for values after the update and the subscript $(\cdot)_{t,t'}$ is used to indicate the unvenly spaced interval between t and t'. The observation matrix $\mathbf{H}_{t,t'}^T$ is obtained by concatenating the bases at the corresponding observation steps, where the Q unobserved states of the robot are represented by zero entries in the diagonal. Thus, for a each time step t,

$$\mathbf{H}_t^T = \left[\begin{array}{ccc:ccc} (\boldsymbol{\psi}_t^T)_1 & \dots & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & (\boldsymbol{\psi}_t^T)_P & \mathbf{0} & \dots & \mathbf{0} \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & \mathbf{0}_{P+1} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0}_{P+Q} \end{array}\right]. \tag{7}$$

Trajectory distributions that predict human and robot movements are obtained by integrating out the weights of the posterior distribution

$$p(\boldsymbol{y}_{0:T}; \boldsymbol{\theta}^{new}) = \int p(\boldsymbol{y}_{0:T} | \bar{\boldsymbol{w}}) p(\bar{\boldsymbol{w}}; \boldsymbol{\theta}^{new}) d\bar{\boldsymbol{w}}. \tag{8}$$

Fig. 2 summarizes the workflow of the Interaction ProMP. During the training phase, imitation learning is used to learn the parameter $\boldsymbol{\theta}$. The distribution is abstracted as a bivariate Gaussian where each of the two dimensions are given by the distribution over the weights of the human and robot trajectories. During execution, the assistive trajectory of the robot is predicted by

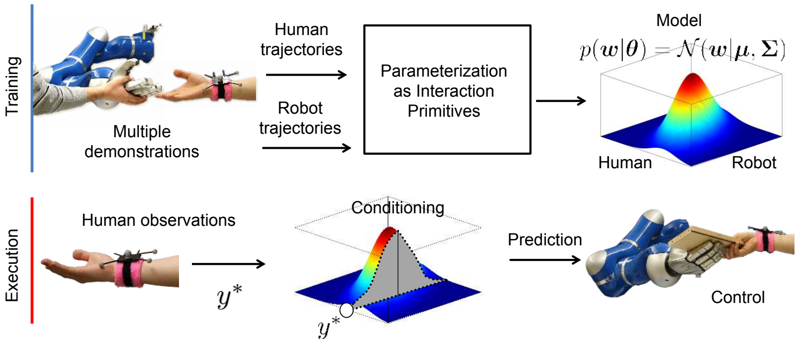

Training
Multiple demonstrations
Human trajectories
Robot trajectories
Parameterization as Interaction Primitives
Model
$p(\boldsymbol{w}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$
Human
Robot
Execution
Human observations
y^*
Conditioning
y^*
Prediction
Control

Fig. 2 The workflow of Interaction ProMP on a single task where the distribution of human-robot parameterized trajectories is abstracted as a bivariate Gaussian. The conditioning step is shown as the slicing of the distribution at the observation of the human position. In the real case, this distribution is multivariate.

integrating out the weights of the posterior distribution $p(\bar{\boldsymbol{w}}; \boldsymbol{\theta}^{new})$. The operation of conditioning is illustrated by the slicing of the prior, at the current observation of the position of the human $\boldsymbol{y}_t^*$.

3.3 Estimating Phases and Actions of Multiple Tasks

ProMPs are learned from multiple demonstrations. Previous works [14, 8] have only addressed spatial variability, but not temporal variability of movements. However, when demonstrating the same task multiple times, a human demonstrator will inevitably execute movements at different speeds, thus changing the phase at which events occur. Previously, this problem has been alleviated by introducing an additional pre-processing step on the training data for time-alignment based on a variant of DTW. Back to Fig. 1(b), the aligned model is shown as the distribution of trajectories indexed by the normalized time.

Time-alignment ensures that the weights of each demonstration can be regressed from the same matrix $\boldsymbol{\psi}_{0:T}$, at the expense that temporal representation is lost within the model. As a consequence, during execution, the conditioning (6) can only be used when the phase of the human demonstrator coincides with the phase encoded by the time-aligned model, which is unrealistic in practice. In [14, 8] we avoided this problem by conditioning only on the last position of the human movement, since for this particular case, the corresponding basis function is known to be $\boldsymbol{\psi}_T$. For any other time step t, the association between $\boldsymbol{y}_t^*$ and the basis $\boldsymbol{\psi}_t$ is unknown given that the human presents temporal variability and that the velocity is either unobserved or computation from derivatives impractical due to sparsity of position measurements[2].

We propose encoding the temporal variance into the model by learning a distribution over phases from the multiple demonstrations. This enriched model not only eliminates the need for time-alignment, but also opens the possibility for faster robot behavior; the conditioning (6) can be applied before the end of the human movement. Initially, we replace the original time indexes of the basis functions with a phase variable $z(t)$. Define T_{nom} as an nominal trajectory duration (e.g. the average final time of the demonstrations) from which the weights of all demonstrations are regressed to obtain the parameters of the distribution $\boldsymbol{\theta} = \{\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w\}$; the Gaussian bases are spread over the nominal duration $\boldsymbol{\psi}_{0:T_{nom}}$. Assuming that each of the i-th demonstrations has a constant rate of change in relation to T_{nom}, define a temporal scaling factor

$$\alpha_i = T_{nom}/T_i. \tag{9}$$

The single scaling factor α_i means that observations (the three red circles in Fig. 1(b)) are "stretched" or "compressed" at the same rate in the temporal direction. Although simple, our experiments have shown that this assumption holds in practice for simple, short stroke movements typical of handovers (see [7] for problems where a multiple-phase algorithm is required). Thus, a trajectory of duration T can be computed relative to the phase

$$p(\boldsymbol{y}_{0:T}|\boldsymbol{w}) = \prod_{0}^{T} \mathcal{N}(\boldsymbol{y}(z_t)|[\boldsymbol{\psi}(z_t)]^T\boldsymbol{w}, \boldsymbol{\Sigma}_y), \qquad z_t = \alpha t. \tag{10}$$

Given the sparse partial sequence of human position observations, a posterior probability distribution over phases is given as

$$p(\alpha|\boldsymbol{y}^*_{t,t'}, \boldsymbol{\theta}) \propto p(\boldsymbol{y}^*_{t,t'}|\alpha, \boldsymbol{\theta})p(\alpha). \tag{11}$$

For simplicity, we assume the prior $p(\alpha)$ as a univariate Gaussian distribution, obtained from the M demonstrations, $\alpha \sim \mathcal{N}(\mu_\alpha, \sigma^2_\alpha)$. For a specific α value the likelihood is

$$\begin{aligned} p(\boldsymbol{y}^*_{t,t'}|\alpha, \boldsymbol{\theta}) &= \int p(\boldsymbol{y}^*_{t,t'}|\bar{\boldsymbol{w}}, \alpha)p(\bar{\boldsymbol{w}})d\bar{\boldsymbol{w}} \\ &= \mathcal{N}(\boldsymbol{y}^*_{t:t'}|[\mathbf{A}(z_{t:t'})]^T\boldsymbol{\mu}_w, [\mathbf{A}(z_{t,t'})]^T\boldsymbol{\Sigma}_w[\mathbf{A}(z_{t,t'})] + \boldsymbol{\Sigma}^*_y), \end{aligned} \tag{12}$$

where

[2] Under a single temporal scaling factor assumption, and in the case direct measurements of velocity were possible, the phase could be directly estimated.

$$[\mathbf{A}(z_{t,t'})]^T = \begin{bmatrix} [\boldsymbol{\psi}(z_{t,t'})]_1^T & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & [\boldsymbol{\psi}(z_{t,t'})]_P^T \end{bmatrix}, \tag{13}$$

is the matrix of basis functions of the observed positions of the human, which corresponds to the observed entries of the full matrix $\mathbf{H}$ in (7), however, now indexed by the phase $z_t = \alpha t$. Given $\boldsymbol{y}_{t,t'}$, (12) is computed for each sampled α candidates, and the most probable scaling value

$$\alpha^* = \arg\max_{\alpha} p(\alpha | \boldsymbol{y}^*_{t,t'}, \boldsymbol{\theta}) \tag{14}$$

is selected. Intuitively, the effect of different phases is to stretch or compress the temporal axis of the prior (unconditioned) distribution proportionally to α. The method then tests which scaling value generates the model with the highest probability given the observation $\boldsymbol{y}^*_{t,t'}$. Once α^* is known, its associated observation matrix $\mathbf{H}(z_{0:T})$ can be used in (6) to condition and predict the trajectories of both human and robot. To efficiently estimate the phase during execution, one approach is to sample a number of values of α from the prior $p(\alpha)$ and precompute and store, for each of them, the associated matrix of basis functions $\mathbf{A}(z_{0:T})$ and $\mathbf{H}(z_{0:T})$ beforehand.

In a multi-task scenario, we can now address the recognition of the task given the positions $\boldsymbol{y}_{t,t'}$. Assume a K number of collaborative tasks are encoded by independently trained Interaction ProMPs represented by the parameter $\boldsymbol{\theta}_k$. For each task k, the most probable α^*_k and likelihood must be stored. By noting that each task is represented by its own parameter $\boldsymbol{\theta}_k$, the most probable task is given by re-using the likelihoods of the set $\{\alpha^*_k, \boldsymbol{\theta}_k\}$. The task recognition is given by

$$k^* = \arg\max_{k} p(k | \boldsymbol{y}^*_{t,t'}), \tag{15}$$

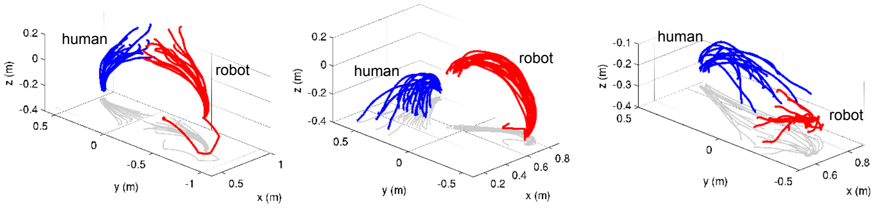
where $p(k|\boldsymbol{y}^*_{t,t'}) \propto p(\boldsymbol{y}^*_{t,t'}|\alpha^*_k, \boldsymbol{\theta}_k)p(k)$ with $p(k)$ being a prior probability distribution of the task and $p(\boldsymbol{y}^*_{t,t'}|\alpha^*_k, \boldsymbol{\theta}_k)$ was previously obtained in (12). The two optimizations in (14) and (15) lead to an algorithm that scales linearly in the number of sampled α's and in the number of tasks.

4 Experiments with a Semi-Autonomous Robot

Collaborative assembly experiments were conducted using a 7-DoF lightweight arm equipped with a 5-finger hand. In all experiments, the wrist of the human was tracked by motion capture, providing XYZ Cartesian coordinates. The trajectories of the robot were recorded by kinesthetic teach-in with the joint encoders. For each demonstration, the set of human-robot measurements were paired and stored with a sampling rate of 50 Hz.

4.1 A Multi-Task Semi-Autonomous Robot Coworker

As it was shown in Fig. 1(a), we applied our method on a multi-task scenario where the robot plays the role of a coworker that helps a human assembling a toolbox. This scenario was previously proposed in [8] where time-alignment was used on the training data. While in our previous work, conditioning could only be computed at the end of the movement, here, the robot can predict the collaborative trajectory before the human finishes moving, leading to a faster robot response. Moreover, since the method described in Section 3 was applied without aligning demonstrations the time spent in pre-processing training data was considerably decreased.



(a) Handing over a plate (b) Handing over a screw (c) Holding the screw driver

Fig. 3 Demonstrations of the three different interactions and their respective trajectories.

The assembly consists of three different collaborative interactions. In one of them, the human extends his hand to receive a plate. The robot fetches a plate from a stand and gives it to the human by bringing it close to his hand. In a second interaction, the human fetches the screwdriver and the robot grasps and gives a screw to the human as a pre-emptive collaborator would do. The third type of interaction consists of the robot receiving a screwdriver such that the human coworker can have both hands free (the same primitive representing this interaction is also used to give the screwdriver back to the human). Each interaction of plate handover, screw handover and holding the screwdriver was demonstrated 15, 20, and 13 times, respectively. The trajectories obtained from the demonstrations are shown in Fig. 3 in the Cartesian space. Note, however, that the interaction primitives were trained on the Cartesian coordinates of the wrist with the joint coordinates of the robot.

To make a direct comparison between the previous method and the present method, the same training data presented in [8] was used. The durations of each demonstration was, however, randomly modified as we noticed the original data did not present sufficient variability of phases (in our initial tests the correct phase could be reasonably estimated with only 2 to 3 α samples).

The randomization acts as a surrogate for different demonstrators with different speeds of execution. The original time-aligned demonstrations for one of the tasks can be seen in Fig. 4(a) as the several gray curves. Using leave-one-out cross-validation (LOOCV) the figure shows that the uncertainty of the human position collapses at the end, when the measurement is made on the test data. The posterior distribution, shown as the blue patch with ± two-standard deviations, predicts the robot final joint positions with an average error of 2.1 ± 6.7 degrees. Fig. 4 (b-c) shows the proposed method with phase-estimation where the training data includes various phases. In Fig. 4(b), the observation represents 25% of the total trajectory length. The final positional error was of 6.8 ± 11.3 degrees. In (c), 50% of trajectory was observed and the error decreased to 2.0 ± 5.7 degrees, roughly achieving the same accuracy as the time-aligned case shown in (a). The error was computed by averaging the RMS final position error over the 7 joint positions of the arm.

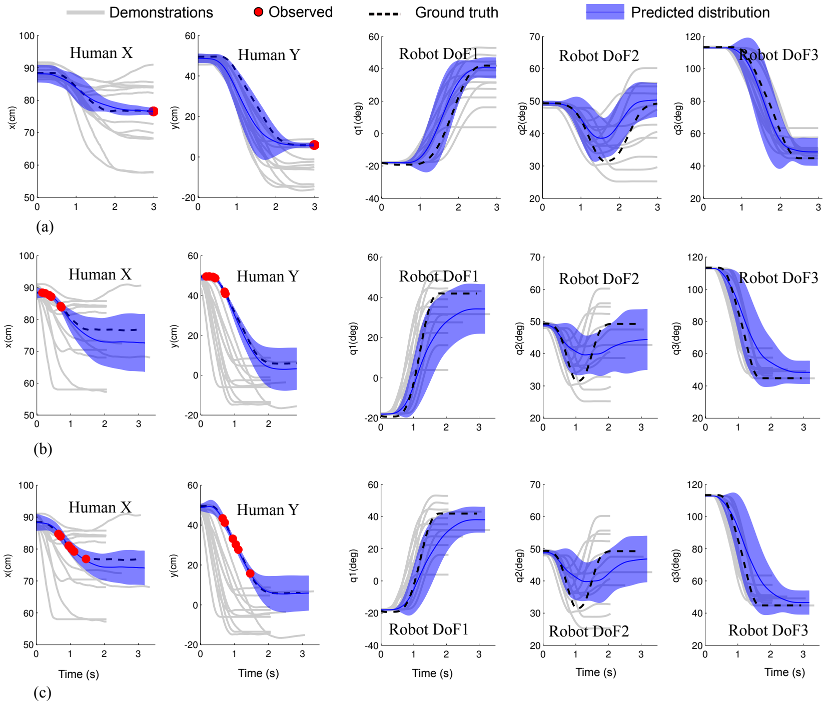
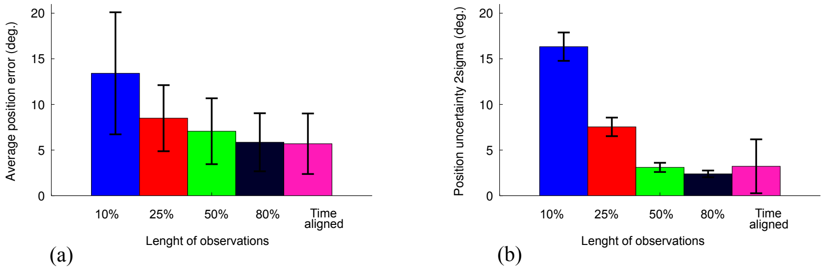


Fig. 4 Prediction of the distribution over trajectories of the human and the robot for the task of handing over a screw driver. (a) The previous method with time-aligned data without phase estimation and therefore conditioned only a the last observation of the human position. The phase estimation method where five measurements randomly spaced are taken up to 25% of the total duration of the human movement in (b) and 50% of the total duration of the human movement in (c).



(a) (b)

Fig. 5 Leave-one-out cross-validation over the whole training dataset of plate handovers. The final position errors (a) and the uncertainty ($\pm 2\sigma$) (b) of the predictions are shown. The predictions are made when 10, 25, 50, and 80% of the trajectory are observed and compared with the time-aligned case.

In Fig. 5(a), each bar represents the final position error as the average over the 7 joints of the robot with LOOCV over the whole data set of demonstrations. The figure shows the cases when 10%, 25%, 50%, and 80% of the trajectory were observed. For each observation, 25 samples of the phase parameter α were used. We compared those results with the original method [8] when the prediction is made based on the final measurement, indicated by the bar labeled "Time aligned". When 80% of the trajectory was observed, the prediction provided the same accuracy of the time-aligned method.

From the same LOOCV test, the uncertainty at the final position (that is, the width of the blue patch previously shown in Fig. 4 at the end of the trajectory), was also quantified. These results are shown at the right plot Fig. 5(b). Note that when 80% of observations were provided, a trajectory with less uncertainty than the time-aligned case can be predicted. This results from the fact that, with phase estimation, a larger number of observations are used to condition the covariance of the prior in (6).

Interaction ProMPs also provide the ability for the robot to spatially coordinate its movement according to the movement of the human. A practical application is the handover of an object at different positions as shown in Fig. 6. In the left picture, the robot first receives the screwdriver from the human. In the right picture, the human extends his hand in a different location and the robot then delivers the tool back. The trajectory of the robot was inferred by conditioning the Interaction Primitive on the first second of the movement of the human. We have previously quantified the accuracy of the prediction in our setup achieving positional errors of less than 3 cm [14].

With phase estimation, the robot reaction time for the handover of the screwdriver shown in Fig. 6 decreased on average by 2 seconds, a reduction of 25% of the task duration in relation to the original time-aligned case. Our preliminary evaluations on the assembly scenario was carried out by sampling 25 values of α phases for each of the three tasks, thus requiring 75 (25 samples $\times$ 3 tasks) calls to the computation of the probabilities (11) while the human moves his arm. The whole process, including the final prediction of the full

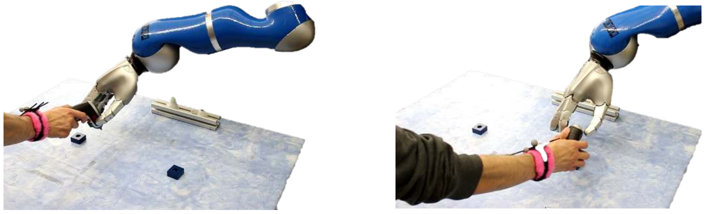


Fig. 6 Handover and return of a screwdriver at different positions, obtained by conditioning the Interaction ProMP on the positions of the wrist marker.

trajectory with (8), observed during the first second of the human movement took in average 0.20 seconds using Matlab code on a conventional laptop (Core i7, 1.7 GHz). To control the robot, only the mean of the posterior distribution over trajectories for each joint of the robot was used, and tracked by the standard, compliant joint controller provided by the robot manufacturer[3]. A video of this experiment can be watched in http://youtu.be/4qDFv02xlNo

4.2 Discussion of the Experiment and Limitations

In practice, there is an upper limit on the number of tasks and sampled α's that can be supported. This limit can be empirically evaluated as the total time required to compute the probability of all sampled alphas, for all tasks, which must be less than the duration of the human movement. Otherwise, it is faster to predict the robot trajectories based on the final measurement of the human, as it was done in previous works. This limit depends on the efficiency of the implementation and the duration of the human movement.

Since the experiments aimed at exposing the adaptation of the robot movement solely by means of the Interaction Primitives, no direct feedback tracking of the marker on the human wrist was made. The Interaction Primitive framework may potentially benefit when used in combination with a feedback controller that tracks the markers directly. Note, however, that it is not possible to completely replace an Interaction Primitive by a controller. A feedback controller does not provide the flexibility and richness of the trajectories that can be encoded in a primitive learned from human demonstrations.

A system that allows for reliable estimation of velocity (or a constant stream of position) can greatly simplify the estimation of the phase, and under the assumption of a constant rate α, make the problem readily solvable. On the other hand, the nondisruptive deployment of semi-autonomous robots in the field must cope with occluded and sparse position measurements, often provided by low-cost sensors such as Kinect cameras, which requires algorithms that are capable of estimating the phase from such data.

[3] Although not used in this paper, the ProMP framework also provides means to compute the feedback controller and the interested reader is referred to [16].

In the short term we envision a self-contained setup that uses a Kinect camera as a replacement of the optical marker tracking system that was used during the experiments.

5 Conclusions

This paper presented a method suited for collaborative and assistive robots whose movements must be coordinated with the trajectories of a human partner moving with different speeds. This goal was achieved by augmenting the previous framework of Interaction ProMPs with a prior model of the phases of the human movement, obtained from demonstrated trajectories. The encoding of phases enriches the model by allowing the alignment of the observations of the human in relation to the interaction model, under an intermittent positional stream of data. We experimentally evaluated our method in an application where the robot acts as a coworker in a factory. Phase estimation allowed our robot to predict the trajectories of both interacting agents before the human finishes the movement, resulting in a faster interaction. The duration of a handover task could be decreased by 25% while using the same robot commands (same speed of the robot movement).

A future application of the method is to use the estimated phase of the human to adapt the velocity of the robot. A slowly moving human suggests that the robot should also move slowly, as an indication that a delicate task is being executed. Conversely, if the human is moving fast, the robot should also move fast as its partner may want to finish the task quickly.

Acknowledgements 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); and from the project BIMROB of the Forum für interdisziplinäre Forschung (FiF) of the TU Darmstadt.

References

1. Ben Amor, H., Neumann, G., Kamthe, S., Kroemer, O., Peters, J.: Interaction primitives for human-robot cooperation tasks. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2014)
2. Bonilla, B.L., Asada, H.H.: A robot on the shoulder: Coordinated human-wearable robot control using coloured petri nets and partial least squares predictions. In: IProceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 119–125 (2014)
3. Calinon, S., Li, Z., Alizadeh, T., Tsagarakis, N.G., Caldwell, D.G.: Statistical dynamical systems for skills acquisition in humanoids. In: Proceedings of the IEEE/RAS International Conference on Humanoids Robots (HUMANOIDS), pp. 323–329 (2012)
4. Calinon, S., Sauser, E.L., Billard, A.G., Caldwell, D.G.: Evaluation of a probabilistic approach to learn and reproduce gestures by imitation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 2671–2676 (2010)

5. Coates, A., Abbeel, P., Ng, A.Y.: Learning for control from multiple demonstrations. In: Proceedings of the 25th International Conference on Machine Learning (ICML), pp. 144–151. ACM (2008)
6. Englert, P., Toussaint, M.: Reactive phase and task space adaptation for robust motion execution. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 109–116 (2014)
7. Ewerton, M., Maeda, G., Peters, J., Neumann, G.: Learning motor skills from partially observed movements executed at different speeds. In: Accepted: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2015)
8. Ewerton, M., Neumann, G., Lioutikov, R., Ben Amor, H., Peters, J., Maeda, G.: Learning multiple collaborative tasks with a mixture of interaction primitives. In: Proceedings of the International Conference on Robotics and Automation (ICRA), pp. 1535–1542 (2015)
9. Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives: learning attractor models for motor behaviors. Neural computation **25**(2), 328–373 (2013)
10. Kim, S., Gribovskaya, E., Billard, A.: Learning motion dynamics to catch a moving object. In: Proceedings of the IEEE/RAS International Conference on Humanoids Robots (HUMANOIDS), pp. 106–111 (2010)
11. Kim, S., Shukla, A., Billard, A.: Catching objects in flight. IEEE Transactions on Robotics (TRO) **30** (2014)
12. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. In: Robotics: Science and Systems (2013)
13. Lee, D., Ott, C., Nakamura, Y.: Mimetic communication model with compliant physical contact in humanhumanoid interaction. The International Journal of Robotics Research **29**(13), 1684–1704 (2010)
14. Maeda, G., Ewerton, M., Lioutikov, R., Ben Amor, H., Peters, J., Neumann, G.: Learning interaction for collaborative tasks with probabilistic movement primitives. In: Proceedings of the International Conference on Humanoid Robots (HUMANOIDS), pp. 527–534 (2014)
15. Mainprice, J., Berenson, D.: Human-robot collaborative manipulation planning using early prediction of human motion. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 299–306. IEEE (2013)
16. Paraschos, A., Daniel, C., Peters, J., Neumann, G.: Probabilistic movement primitives. In: Advances in Neural Information Processing Systems (NIPS), pp. 2616–2624 (2013)
17. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on **26**(1), 43–49 (1978)
18. Schaal, S.: Is imitation learning the route to humanoid robots? Trends in cognitive sciences **3**(6), 233–242 (1999)
19. Tanaka, Y., Kinugawa, J., Sugahara, Y., Kosuge, K.: Motion planning with worker's trajectory prediction for assembly task partner robot. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1525–1532. IEEE (2012)
20. Van Den Berg, J., Miller, S., Duckworth, D., Hu, H., Wan, A., Fu, X., Goldberg, K., Abbeel, P.: Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 2074–2081 (2010)
21. Vuga, R., Nemec, B., Ude, A.: Speed profile optimization through directed explorative learning. In: Proceedings of the IEEE/RAS International Conference on Humanoids Robots (HUMANOIDS), pp. 547–553. IEEE (2014)
22. Yamane, K., Revfi, M., Asfour, T.: Synthesizing object receiving motions of humanoid robots with human motion database. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 1629–1636. IEEE (2013)