Model-Free Probabilistic Movement Primitives for Physical Interaction

Alexandros Paraschos[1], Elmar Rueckert[1], Jan Peters[1,2] and Gerhard Neumann[1]

***Abstract*— Physical interaction in robotics is a complex problem that requires not only accurate reproduction of the kinematic trajectories but also of the forces and torques exhibited during the movement. We base our approach on Movement Primitives (MP), as MPs provide a framework for modelling complex movements and introduce useful operations on the movements, such as generalization to novel situations, time scaling, and others. Usually, MPs are trained with imitation learning, where an expert demonstrates the trajectories. However, MPs used in physical interaction either require additional learning approaches, e.g., reinforcement learning, or are based on handcrafted solutions. Our goal is to learn and generate movements for physical interaction that are learned with imitation learning, from a small set of demonstrated trajectories. The Probabilistic Movement Primitives (ProMPs) framework is a recent MP approach that introduces beneficial properties, such as combination and blending of MPs, and represents the correlations present in the movement. The ProMPs provides a variable stiffness controller that reproduces the movement but it requires a dynamics model of the system. Learning such a model is not a trivial task, and, therefore, we introduce the model-free ProMPs, that are learning jointly the movement and the necessary actions from a few demonstrations. We derive a variable stiffness controller analytically. We further extent the ProMPs to include force and torque signals, necessary for physical interaction. We evaluate our approach in simulated and real robot tasks.**

I. Introduction

Developing robots that can operate in the same environment with humans and physically interacting with every-day objects requires accurate control of the contact forces that occur during the interaction. While non-compliant robots can achieve a great accuracy, the uncertainty of complex and less-structured environment prohibits physical interaction. In this paper, we focus on providing a compliant control scheme that can enable robots to manipulate their environment without damaging it. Typically, force-control requires an accurate dynamics model of the robot and its environment that is not easy to obtain. Other approaches suggest to learn a dynamics model, however, this process can be time-consuming and is prone to model-errors. We present an approach that can jointly learn the desired movement of the robot and the contact forces by human demonstrations, without relying on a learned forward or inverse model.

*The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreements #600716 (CoDyCo) and #270327 (CompLACS)

[1]Intelligent Autonomous Systems, TU Darmstadt, 64289 Darmstadt, Germany {paraschos,neumann,rueckert}@ias.tu-darmstadt.de

[2] Robot Learning Group, Max Planck Institute for Intelligent Systems, Germany mail@jan-peters.net

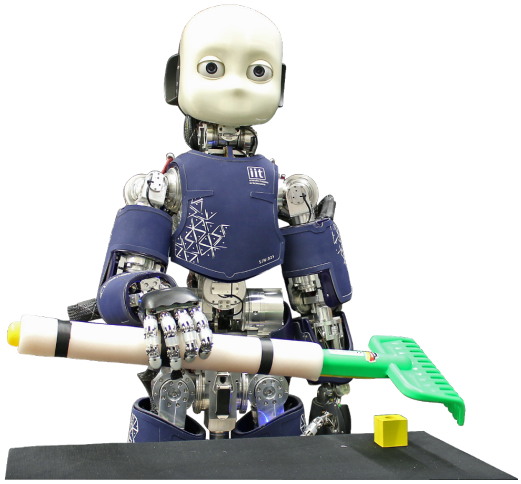

Fig. 1. The *iCub* robot is taught by imitation how to tilt a grate that we use of during the experimental evaluation of our approach. We demonstrated how to lift a grate from three different positions. Grasping from different positions change the dynamics of the task. Our method provides online adaptation and generalizes in the area of the grasps

Existing approaches for motor skill learning that are based on movement primitives [1], [2], [3], [4], [5], often incorporate into the movement primitive representation the forces needed for the physical interactions [6], [7], [8]. However, such approaches model a single successful reproduction of the task. Multiple demonstrations are typically averaged, despite that they actually represent similar, but different, solutions of the task. Thus, the applied contact forces are not correlated with the state of the robot nor sensory values that indicate the state of the environment, e.g. how heavy an object is.

In this paper, we propose learning the coordination of the interaction forces, with the kinematic state of the system, as well as the control actions needed to reproduce the movement exclusively from demonstration. Motor skill learning for such interaction tasks for high-dimensional redundant robots is challenging. This task requires real-time feedback control laws that process sensory data including joint encoders, tactile feedback and force-torque readings. We present a model-free version of the Probabilistic Movement Primitives (ProMPs) [9] that enables robots to acquire complex motor skills from demonstrations, while it can coordinate the movement with force, torque, or tactile sensing. The ProMPs have several beneficial properties, such us generalization to novel situations, combination of primitives and time-scaling, which we inherit in our approach.

ProMPs assume a locally linearizable dynamics models to

compute time-varying feedback control laws. However, such dynamics models are hard to obtain for physical interaction tasks. Therefore we obtain a time varying feedback controller directly from the demonstration without requiring such a model. In the model-free extension of the ProMPs, we condition the joint distribution over states and controls on the current state of the system, and obtain a distribution over the controls. We show that this distribution represents a time-varying stochastic linear feedback controller. Due to the time-varying feedback gains, the controller can exhibit behavior with variable stiffness and, thus, it is safe to use in physical interaction. A similar control approach has recently been presented in [10].

Our approach inherits many beneficial properties of the original ProMP formulation. We can reproduce the variability in the demonstrations and use probabilistic operators for generalization to new tasks or the co-activation of learned primitives. The resulting feedback controller shows similar properties as in the model-based ProMP approach, it can reproduce optimal behavior for stochastic systems and exactly follow the learned trajectory distribution, at least, if the real system dynamics are approximately linear for each time step. For non-linear systems, the estimated variable stiffness controller can get unstable if the robot reaches configurations that are far away from the set of demonstrations. To avoid this problem, we smoothly switch between a stable PD-controller and the ProMP controller if the support of the learned distribution for the current situation is small. We show that this extension allows us to track trajectory distributions accurately even for non-linear systems.

The model-free ProMP approach is evaluated in simulation with linear and non-linear dynamical systems in Section V. In a real task, we tilt a grate which is grasped at different positions. We show that the model-free ProMPs can generalize to different grasping locations on a grate through exploiting the correlations between motor commands and force feedback.

II. Related Work

In this section, we review related work on movement primitives for imitation learning that combine position and force tracking, model the coupling between kinematics and forces and are able to capture the correlations between these two quantities.

The benefit of an additional feedback controller to track desired reference forces was demonstrated in grasping tasks in [6]. Individual dynamical systems (DMPs) [5] were trained for both, position and force profiles in imitation learning. The force feedback controller substantially improved the success rate of grasps in tracking demonstrated contact forces under changing conditions. For manipulation tasks like opening a door, the authors showed that the learned force profiles can be further improved through reinforcement learning [7].

For many tasks, such as like bi-manual manipulations, the feedback controller needs to be coupled. Gams et al. [11] proposed *cooperative* dynamical systems, where deviations from desired forces modulate the velocity forcing term in the DMPs for position control. This approach was tested on two independently operating robot arms solving cooperative tasks like lifting a stick [8]. Deviations in the sensed contact forces in one robot were used to adapt the DMP of the other robot and the coupling parameters were obtained through iterative learning control. A related probabilistic imitation learning approach to capture the couplings in time was proposed in [12]. In this approach, Gaussian mixture models were used to represent the variance of the demonstrations. The approach was evaluated successfully on complex physical interaction tasks such as ironing, opening a door, or pushing against a wall.

Adapting Gaussian Mixture Models (GMMs) [13], [14], [15], [16] have been proposed for use in physical interaction tasks. The major difference to the dynamical systems approach is that GMMs can represent the variance of the movement. Closely related to our approach, Evrard et al. in [17] used GMMs to learn joint distributions of positions and forces. Joint distributions capture the correlations between positions and forces and were used to improve adaptation to perturbations in cooperative human robot tasks for object lifting. In this approach, the control gains were fixed to track the mean of the demonstrated trajectories. In [18], it was shown that by assuming known forward dynamics, variable stiffness control gains can be derived in closed form to match the demonstrations. We address here an important related question of how these gains can be learned in a model-free approach from the demonstrations.

III. Model-Free Probabilistic Movement Primitives

We propose a novel framework for robot control which can be employed in physical interaction scenarios. In our approach, we jointly learn the desired trajectory distribution of the robot's joints or end-effectors and the corresponding controls signals. We train our approach from a limited set of demonstrations. We refer to the joint distribution as state-action distribution. Further, we incorporate proprioceptive sensing, such as force or tactile sensing, into our state representation. The additional sensing capabilities are of high importance for physical interaction as they can disambiguate kinetically similar states. We present our approach by, first, extending the Probabilistic Movement Primitives (ProMPs) framework [9] to encode the state-action distribution and, second, we derive a stochastic feedback controller without the use of a given system dynamics model. Finally, we extend our control approach for states which are relatively far from the vicinity of the learned state-action distribution. In that case, our control approach can no longer produce correcting actions, and an additional backup controller with high gains is needed. Our framework inherits most of the beneficial properties introduced by the ProMPs that significantly improved generalization to novel situations and enables the generation of primitives that *concurrently* solve multiple tasks [9].

A. Encoding the Time-Varying State-Action Distribution of the Movement

We avoid explicitly learning robot and environment models by learning directly the appropriate control inputs, while keeping the beneficial properties of the ProMP approach, such us generalization and concurrent execution.

In order to simplify the illustration of our approach, we first discuss the special case of a single Degree of Freedom (DoF) and, subsequently, we expand our description to the generic case of multiple DoF. The description is based in [9], but modified appropriately to clarify how the actions can be modelled. First, we define the extended state of the system as

$$\boldsymbol{y}_t = [q_t, \dot{q}_t, u_t]^T, \tag{1}$$

where q_t is the position of the joint, $\dot{q}_t$ the velocity, and u_t the control applied at time-step t. Similar to ProMPs, we use a linear basis function model to encode the trajectory of the extended state $\boldsymbol{y}_t$. The feature matrix and the weight vector of the non-linear function approximation model become

$$\boldsymbol{y}_t = \begin{bmatrix} q_t \\ \dot{q}_t \\ u_t \end{bmatrix} = \tilde{\boldsymbol{\Phi}}_t \boldsymbol{w}, \quad \tilde{\boldsymbol{\Phi}}_t = \begin{bmatrix} \boldsymbol{\phi}_t^T & \mathbf{0} \\ \dot{\boldsymbol{\phi}}_t^T & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\psi}_t^T \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_q \\ \boldsymbol{w}_u \end{bmatrix}, \tag{2}$$

where the vectors $\boldsymbol{\phi}_t$ and $\boldsymbol{\psi}_t$ represent the feature vectors for the position q_t and the control u_t respectively. The derivative of the position feature vector $\dot{\boldsymbol{\phi}}_t$ is used to compute the velocity of the joint $\dot{q}_t$. The weight vector $\boldsymbol{w}$ contains the weight vector for the position $\boldsymbol{w}_q$ and the weight vector for the control $\boldsymbol{w}_u$. The dimensionality of the feature $\boldsymbol{\phi}_t$ and weight $\boldsymbol{w}_q$ vectors is $N \times 1$, where N is the number of features used to encode the joint position. Similarly, the dimensionality of $\boldsymbol{\psi}_t$ and $\boldsymbol{w}_u$ vectors is $M \times 1$. The remaining entries of $\tilde{\boldsymbol{\Phi}}_t$, denoted by $\mathbf{0}$, are zero-matrices with the appropriate dimensionality. In our approach, we distinct between the features used to encode the position from the features used to encode the control signal due to the different properties of the two signals. The distinction allows us to use of different type of basis functions, different parameters, or a different number of basis functions.

We extend our description to the multidimensional case. First, we extend the state of the system from Equation (2) to

$$\boldsymbol{y}_t = \left[\boldsymbol{q}_t^T, \dot{\boldsymbol{q}}_t^T, \boldsymbol{u}_t^T\right]^T, \tag{3}$$

where the vector $\boldsymbol{q}_t$ is a concatenation of the positions of all joints of the robot, the vector $\dot{\boldsymbol{q}}_t$ of the velocities of the joints, and $\boldsymbol{u}_t$ of the controls respectively. The feature matrix $\tilde{\boldsymbol{\Phi}}_t$ now becomes a block matrix

$$\tilde{\boldsymbol{\Phi}}_t = \left[\boldsymbol{\Phi}_t^T, \dot{\boldsymbol{\Phi}}_t^T, \boldsymbol{\Psi}_t^T\right]^T, \tag{4}$$

where

$$\boldsymbol{\Phi}_t = \left[\begin{array}{ccc|c} \boldsymbol{\phi}_t^T & \cdots & \mathbf{0} & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ \mathbf{0} & \cdots & \boldsymbol{\phi}_t^T & \end{array}\right], \tag{5}$$

$$\boldsymbol{\Psi}_t = \left[\begin{array}{c|ccc} & \boldsymbol{\psi}_t^T & \cdots & \mathbf{0} \\ \mathbf{0} & \vdots & \ddots & \vdots \\ & \mathbf{0} & \cdots & \boldsymbol{\psi}_t^T \end{array}\right], \tag{6}$$

define the features for the joint positions and the joint controls. Similarly to the single DoF, the features used for the joint velocities $\dot{\boldsymbol{\Phi}}_t$ are the time derivatives of the features of the joint positions $\boldsymbol{\Phi}_t$. We use the same features for every DoF. The dimensionality of the feature matrices $\boldsymbol{\Phi}_t$ and $\boldsymbol{\Psi}_t$ is $K \times K \cdot (N + M)$, where K denotes the number of DoF.

The weight vector $\boldsymbol{w}$ has a similar structure to Equation (2) and, for the multi-DoF case, is given by

$$\boldsymbol{w} = \Big[\underbrace{{}^1\boldsymbol{w}_q^T, \cdots, {}^K\boldsymbol{w}_q^T}_{\text{weights for joint positions}}, \underbrace{{}^1\boldsymbol{w}_u^T, \cdots, {}^K\boldsymbol{w}_u^T}_{\text{weights for joint controls}} \Big]^T, \tag{7}$$

where ${}^i\boldsymbol{w}$ denotes the weight vector for joint $i \in [1, K]$.

The probability of a single trajectory $\boldsymbol{\tau} = \{\boldsymbol{y}_t, t \in [1 \cdots T]\}$, composed from states of T subsequent time steps, given the parameters $\boldsymbol{w}$, is computed by

$$p(\boldsymbol{\tau}|\boldsymbol{w}) = \prod_t \mathcal{N}\left(\boldsymbol{y}_t | \boldsymbol{\Phi}_t \boldsymbol{w}, \boldsymbol{\Sigma}_{\boldsymbol{y}}\right), \tag{8}$$

where we assume *i.i.d.* Gaussian observation noise with zero mean and $\boldsymbol{\Sigma}_{\boldsymbol{y}}$ covariance. Representing multiple trajectories would require a set of weights $\{\boldsymbol{w}\}$. Instead of explicitly maintaining such a set, we introduce a distribution over the weights $p(\boldsymbol{w}; \boldsymbol{\theta})$, where the parameter vector $\boldsymbol{\theta}$ defines the parameters of the distribution. Given the distribution parameters $\boldsymbol{\theta}$, the probability of the trajectory becomes

$$p(\boldsymbol{\tau}; \boldsymbol{\theta}) = \int p(\boldsymbol{\tau}|\boldsymbol{w}) p(\boldsymbol{w}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{w}, \tag{9}$$

where we marginalize over the weights $\boldsymbol{w}$. As in the ProMP approach, we use a Gaussian distribution to represent $p(\boldsymbol{w}; \boldsymbol{\theta})$, where $\boldsymbol{\theta} = \{\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}\}$. Using a Gaussian distribution enables the marginal to be computed analytically and facilitates learning. The distribution over the weight vector $p(\boldsymbol{w}; \boldsymbol{\theta})$ correlates (couples) the DoFs of the robot to the action vector at every time-step t. The probability of the current state-action vector $\boldsymbol{y}_t$ given $\boldsymbol{\theta}$ is computed by

$$\begin{aligned} p(\boldsymbol{y}_t; \boldsymbol{\theta}) &= \int \mathcal{N}\left(\boldsymbol{y}_t | \boldsymbol{\Phi}_t \boldsymbol{w}, \boldsymbol{\Sigma}_{\boldsymbol{y}}\right) \mathcal{N}\left(\boldsymbol{w} | \boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}\right) \mathrm{d}\boldsymbol{w} \\ &= \mathcal{N}\left(\boldsymbol{y}_t \middle| \boldsymbol{\Phi}_t \boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Phi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Phi}_t^T + \boldsymbol{\Sigma}_{\boldsymbol{y}}\right), \end{aligned} \tag{10}$$

in closed form. We use normalized Gaussian basis functions as features. Each basis function is defined in the time domain by

$$\phi_i(t) = \frac{b_i(t)}{\sum_{j=1}^n b_j(t)}, \tag{11}$$

$$b_i(t) = \exp\left(-\frac{(t - c_i)^2}{2h}\right), \tag{12}$$

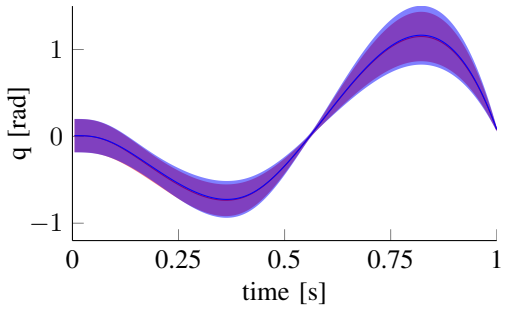

Fig. 2. We evaluate our approach on a simulated 1-DoF linear system. We use $N = 30$ demonstrations (red) for training. During the reproduction (blue) our approach matches exactly the demonstrations.

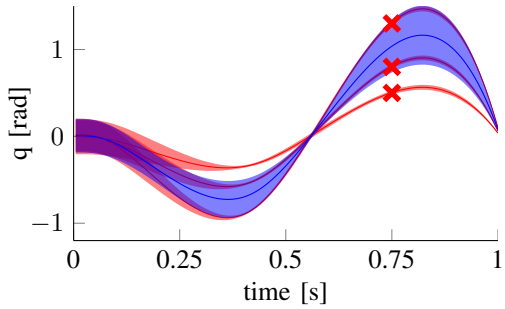

Fig. 3. We evaluate the generalization capabilities of our approach with *conditioning*. The initial distribution is depicted in blue. At time $t = 0.75s$ we condition the initial distribution to pass at a specific position $q = \{0.5, 0.8, 1.3\}$ with low variance. We generate $N = 30$ demonstrations for every conditioning point and we show the resulting distribution in red. The *X* markers denotes the position at the conditioning point.

where c_i denotes the center of the ith basis function and h the bandwidth. The centers of the basis functions are spread uniformly in $[-2h, T_{\text{end}}+2h]$. The number of basis functions and the bandwidth value we used, depend on the complexity of task. Typically, complex task require higher number of basis functions in order to represent them accurately.

B. Imitation Learning for Model-Free ProMPs

We use multiple demonstrations to estimate the parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}\}$ of the distribution over the weights $p(\boldsymbol{w}|\boldsymbol{\theta})$. First, for each demonstration i, we use linear ridge regression to estimate the parameter vector $\boldsymbol{w}_i$ associated to that specific demonstration, i.e.,

$$\boldsymbol{w}_i = (\boldsymbol{\Phi}_t^T \boldsymbol{\Phi}_t + \lambda \boldsymbol{I})^{-1} \boldsymbol{\Phi}_t^T \boldsymbol{Y}_i, \tag{13}$$

where λ denotes the ridge factor and $\boldsymbol{Y}_i$ the observations of the state and action for all the time steps of that demonstration. We set λ to zero, unless numerical issues arise. Subsequently, we estimate the parameters $\boldsymbol{\theta}$ from the set of weights $\{\boldsymbol{w}_i, i \in [1, N]\}$ using the ML estimators for Gaussians, i.e.,

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{w}} &= \frac{1}{L} \sum_{i=1}^{L} \boldsymbol{w}_i, \\ \boldsymbol{\Sigma}_{\boldsymbol{w}} &= \frac{1}{L} \sum_{i=1}^{L} (\boldsymbol{w}_i - \boldsymbol{\mu}_{\boldsymbol{w}})(\boldsymbol{w}_i - \boldsymbol{\mu}_{\boldsymbol{w}})^T, \end{aligned} \tag{14}$$

where L is the number of demonstrations.

C. Integration of Proprioceptive Feedback

Additional sensory feedback integration, e.g., force-torque feedback, is beneficial for physical interaction scenarios as we can capture the correlation of the trajectory, the controls and the sensory signal. This correlation might contain useful information for the reproduction of the movement. We extend our approach to additionally contain the sensory signal $\boldsymbol{s}_t$. The extension require the state $\boldsymbol{y}_t$ to include the sensory signal $\boldsymbol{s}_t$. We estimate an individual weight vector $\boldsymbol{w}_s$ that we include in the concatenated weight vector $\boldsymbol{w}$. Hence, by learning the distribution $p(\boldsymbol{w})$, we can represent the correlations between the sensory signal and the control commands. We use the sensory signal to get a new desired trajectory distribution and its controls.

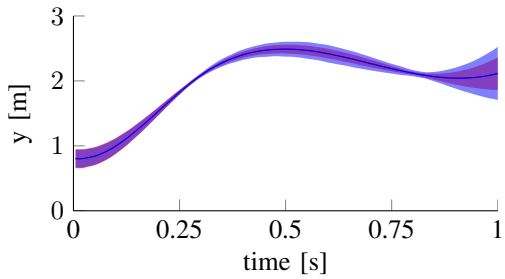

Fig. 4. We evaluate our approach on an non-linear system with $D = 4$ DoF. While the dynamics of the task are non-linear we are able to reproduce (blue) accurately the demonstrated distribution (red). We show the trajectory distribution of the "y" dimension of the task-space of the robot. Our approach captures the correlations between the DoF of the robot and reduces the variance of the trajectory reproduction at both via-points.

D. Generalization with Conditioning

The modulation of via-points and final positions is an important property of any MP framework to adapt to new situations. Generalization to different via-points or final targets can be implemented by conditioning the distribution at reaching the desired position $\boldsymbol{q}_t^*$ (or by conditioning on any other sensory value) at time step t.

By applying Bayes theorem, we obtain a new distribution $p(\boldsymbol{w}|\boldsymbol{q}_t^*)$ for $\boldsymbol{w}$ which is Gaussian with mean and variance

$$\boldsymbol{\mu}_{\boldsymbol{w}}^{[\text{new}]} = \boldsymbol{\mu}_{\boldsymbol{w}} + \boldsymbol{Q}_t \left(\boldsymbol{q}_t^* - \boldsymbol{\Psi}_t^T \boldsymbol{\mu}_{\boldsymbol{w}} \right), \tag{15}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{w}}^{[\text{new}]} = \boldsymbol{\Sigma}_{\boldsymbol{w}} - \boldsymbol{Q}_t \boldsymbol{\Psi}_t^T \boldsymbol{\Sigma}_{\boldsymbol{w}}, \tag{16}$$

$$\boldsymbol{Q}_t = \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_t \left(\boldsymbol{\Sigma}_q^* + \boldsymbol{\Psi}_t^T \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_t \right)^{-1}, \tag{17}$$

where $\boldsymbol{\Sigma}_q^*$ is a covariance matrix specifying the accuracy of the conditioning. As the weight vectors for the controls are also contained in the distribution, the distribution over the controls will change accordingly, such that, by executing the controls, we will reach the desired state $\boldsymbol{q}_t^*$.

E. Robot Control with Model-Free ProMPs

We derive a stochastic feedback controller which is ideally capable of reproducing the learned distribution. We define as $\tilde{\boldsymbol{y}}_t$ the observable state of the system, that contains the joint

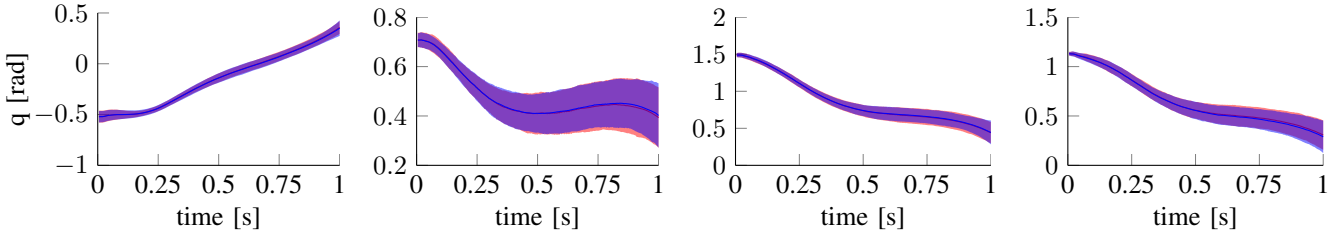

Fig. 5. The evaluation of our approach on the quad-link robot. We present the results of the of the DoF in joint space. The demonstrated distribution is plotted in red and the reproduction in blue. The two distributions match. The two via-points of the movement, which were set in task-space, are not visible in joint-space.

positions, velocities, and potentially force or torque data, but not the action. We rewrite the joint probability

$$p(\boldsymbol{y}_t) = p(\tilde{\boldsymbol{y}}_t, \boldsymbol{u}_t) = \mathcal{N}\left(\begin{bmatrix}\tilde{\boldsymbol{y}}_t \\ \boldsymbol{u}_t\end{bmatrix} \middle| \tilde{\boldsymbol{\Phi}}_t \boldsymbol{\mu}_{\boldsymbol{w}}, \tilde{\boldsymbol{\Phi}}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \tilde{\boldsymbol{\Phi}}_t^T + \boldsymbol{\Sigma}_{\boldsymbol{y}}\right), \quad (18)$$

where

$$\tilde{\boldsymbol{\Phi}}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \tilde{\boldsymbol{\Phi}}_t^T = \begin{bmatrix}\boldsymbol{\Phi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Phi}_t^T & \boldsymbol{\Phi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_t^T \\ \boldsymbol{\Psi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Phi}_t^T & \boldsymbol{\Psi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_t^T\end{bmatrix}, \quad (19)$$

and condition on the current observable state $\tilde{\boldsymbol{y}}_t$ to obtain the desired action. From the Bayes theorem, we obtain the probability of the desired action

$$p(\boldsymbol{u}_t | \tilde{\boldsymbol{y}}_t) = \frac{p(\tilde{\boldsymbol{y}}_t, \boldsymbol{u}_t)}{p(\tilde{\boldsymbol{y}}_t)} = \mathcal{N}\left(\boldsymbol{u}_t | \boldsymbol{\mu}_{\boldsymbol{u}}, \boldsymbol{\Sigma}_{\boldsymbol{u}}\right), \quad (20)$$

which is a Gaussian distribution as both $p(\tilde{\boldsymbol{y}}_t)$ and $p(\boldsymbol{u}_t)$ are Gaussian. The mean and covariance of $p(\boldsymbol{u}_t)$ are computed by

$$\boldsymbol{\mu}_{\boldsymbol{u}} = \boldsymbol{\Psi}_t \boldsymbol{\mu}_{\boldsymbol{w}} + \boldsymbol{K}_t, (\tilde{\boldsymbol{y}}_t - \boldsymbol{\Phi}_t \boldsymbol{\mu}_{\boldsymbol{w}}) \quad (21)$$

$$\boldsymbol{\Sigma}_{\boldsymbol{u}} = \boldsymbol{\Psi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_t^T + \boldsymbol{K}_t \, \boldsymbol{\Phi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Phi}_t^T, \quad (22)$$

$$\boldsymbol{K}_t = \boldsymbol{\Psi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Phi}_t^T \left(\boldsymbol{\Phi}_t \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Phi}_t^T\right)^{-1}, \quad (23)$$

using Gaussian identities. We rewrite the mean control given the observable state $\tilde{\boldsymbol{y}}_t$ as

$$\boldsymbol{\mu}_{\boldsymbol{u}} = \boldsymbol{\Psi}_t \boldsymbol{\mu}_{\boldsymbol{w}} + \boldsymbol{K}_t \tilde{\boldsymbol{y}}_t - \boldsymbol{K}_t \boldsymbol{\Phi}_t \boldsymbol{\mu}_{\boldsymbol{w}} = \boldsymbol{K}_t \tilde{\boldsymbol{y}}_t + \boldsymbol{k}_t, \quad (24)$$

and observe that it has the same structure as a feedback controller with time varying gains. The feedback gain matrix $\boldsymbol{K}_t$ couples the DoF and the additional force-torque signals of the system. The control covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{u}}$ introduces correlated noise in the controls. The noise used only if we want to match the variability of the demonstrations. Alternatively, we can disable the noise and replay the noise-free behavior.

F. Correction Terms for Non-Linear Systems

A basic assumption for the linear feedback controller obtained by the ProMP approach is that the movement is defined in a local vicinity such that a linear controller is sufficient. Whenever the robot's state "leaves" this vicinity, due to the non-linearities of the dynamics, the learned feedback controller might not be able to direct the robot back to the desired trajectory distribution. Therefore, we apply a correction controller that is active only when the state is sufficiently "far" outside the distribution and directs the system to the mean of the demonstrated state distribution. The correction controller is defined as a standard PD controller with hand-tuned gains, i.e.,

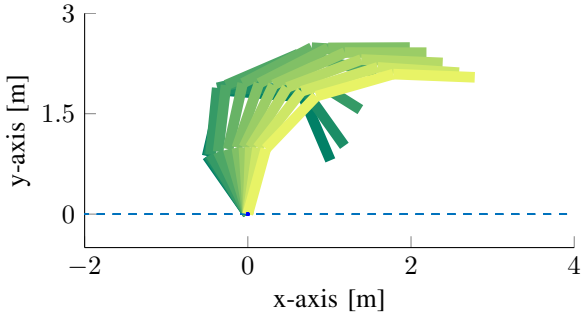

Fig. 6. An animation of the movement of the quad-link non-linear robot during the execution of our approach. We use darker colors at the beginning of the movement and lighter at the end.

$$\boldsymbol{u}_t^C = \boldsymbol{K}_P \left(\boldsymbol{\mu}_{q,t} - \boldsymbol{q}_t\right) + \boldsymbol{K}_D \left(\boldsymbol{\mu}_{\dot{q},t} - \dot{\boldsymbol{q}}\right) + \boldsymbol{u}_{\text{ff},t}, \quad (25)$$

where the feed forward term $\boldsymbol{u}_{\text{ff},t}$ is still estimated from the ProMP and given by the mean action of the ProMP for time step t, i.e.,

$$\boldsymbol{u}_{\text{ff},t} = \boldsymbol{K}_t \boldsymbol{\Phi}_t \boldsymbol{\mu}_{\boldsymbol{w}} + \boldsymbol{k}_t. \quad (26)$$

The correcting action $\boldsymbol{u}_t^C$ is only applied if we are outside the given trajectory distribution. We use a sigmoid activation function that depends on the log-likelihood of the current state to switch between the ProMP feedback controller and the correction controller,

$$\sigma\left(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t\right) = \frac{1}{1 + \exp\left(-\log\left(p\left(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t; \boldsymbol{\theta}\right)\right) \beta^{-1} - \alpha\right)}, \quad (27)$$

where α and β are hand tuned parameters of the activation function. We linearly interpolate between the controls of the ProMP and the correction action. For a high likelihood, e.g., $\sigma(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t) = 1$ we fully activate the feedback controller from the ProMP. For $\sigma(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t) = 0$ we fully activate the correction action.

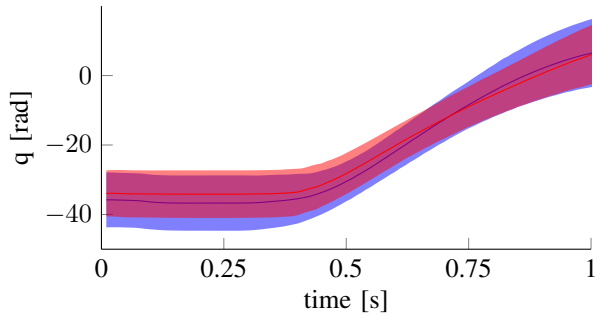

Fig. 7. The trajectory distribution of the wrist joint of the iCub during our experiment. The demonstrated distribution is presented in blue and the reproduction in red. The demonstrated distribution contain trajectories from all three grasping locations. The reproduction distribution contain trajectories from seven grasping locations. The Model-Free ProMPs can reproduce the demonstrated distribution in new grasping locations.

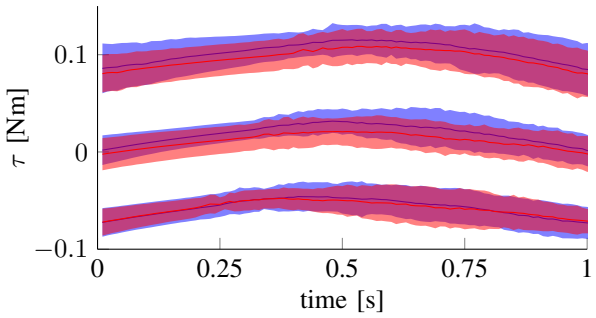

Fig. 8. The torque distribution of all grasping locations used during the demonstrations. Each location created a distinct offset in the measured torque. We present the demonstrated torque distributions in blue. Additionally, we show that our approach can reproduce the torque distribution when we position the grate at the same locations as in the demonstrations. We present the reproduction results in red.

IV. Experimental Evaluation

We begin our experimental evaluation with different toy tasks to demonstrate the properties of the model-free ProMP approach. First, we demonstrate that our model-free ProMP controller can reproduce the demonstrated trajectory distribution accurately on a linear 1-D system. Then, we change the desired trajectory distribution by conditioning, to generalize to different via-points and we execute our controller. We shot that the resulting distribution exactly reaches the via-points.

In a sub-sequent experiment, we test the model-free ProMP on a non-linear Quad-Link pendulum. We show that by the use of the correcting PD controller we can still track the distribution accurately.

Finally, we performed first experiments on the iCub, where the humanoid is grasping a grate at different grasp locations and has to tilt it. By learning the correlation between the force-torque readings and the demonstrated control actions the iCub should learn to compensate for gravitational effects.

A. Reproduction of the Trajectory Distribution

We illustrate our approach in an one dimensional linear second order integrator as the underlying dynamical system. We created the demonstrations by first creating different desired trajectories with splines that go through different via-points. The real trajectories are created by following a given spline with a PD control law. We also added noise to the acceleration of the system. The resulting trajectory distribution is given in Fig. 2 (red). In the same figure, we illustrated the resulting trajectory distribution by using the ProMP controller from the learned model-free ProMP. As we can see, the controller could match the distribution accurately.

B. Generalization by Conditioning to Different Via-Points

We test the conditioning operations that can be performed upon the trajectory distribution to generalize to different via-points. We conditioned the trajectory distribution to reach the positions $0.5, 0.8$ and 1.3 respectively at time point $t = 0.75s$. The resulting via points are indicated by a red cross in Fig. 3. For each of the conditioning scenarios, we plot the resulting trajectory distribution when executing the controller of the conditioned ProMP. The model-free ProMP controller keeps the shape of the distribution while reaching the desired via-points.

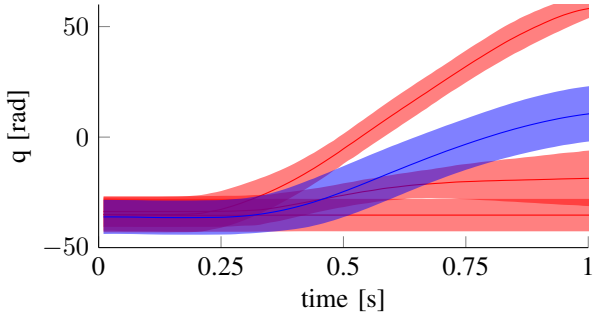

Fig. 9. The trajectory distribution of the wrist joint of the robot, when we disable the torque feedback. Depending on the grasping location, the robot either fails to lift the grate to the same height as demonstrated, or, it overshoots the lifting task due to gravity. In the later case, it should be noted that the center of mass of the grate is moved over the axis of the joint and, thus, gravity forces the grate to lift. For comparison, we present the demonstration distribution from all grasping locations in blue.

C. Non-Linear Quad-Link Pendulum

To evaluate the quality of our controller on a non-linear system, we tested our model-free ProMP approach on a non-linear quad-link planar pendulum. Each link had a mass of 1kg and a length of 1m. We used the standard rigid body dynamics equations, where the gravity and the Coriolis forces are the major non-linear terms. We collected demonstrations by defining the desired trajectory as a spline with two via-point at $t = 0.3, 0.8$ in the task-space of the robot. We generated the demonstration trajectories using inverse kinematics for generating the joint space reference trajectories. Then, we used a inverse dynamics controller to track the reference trajectories and we collected the joint state-action data. We trained our approach using $N = 30$ demonstrations.

The resulting trajectory distribution for the y-dimension of the task-space is show in Fig. 4. The robot can track with its end-effector the desired distribution accurately and can reproduce the two via-points. In Fig. 5 we show all four

the joint trajectories. In the joint space distributions the via-points are not visible but are captured in the covariance matrix of the weights. While the distribution is a wide, the controller could match the mean and variance of the demonstrated trajectory distribution. In Fig. 6, we illustrated the resulting trajectory from the controller in the task space of the robot. The activation of the correcting controller is around 1% of the total execution time.

D. Adaptation to External Forces on the ICub

In this experiment we used the presented model-free ProMP approach to learn a one-dimensional torque feedback controller in the humanoid robot iCub. The task is to tilt a grate multiple times from an initial distribution to a goal distribution, as show in Fig. 7. In our experiments we use the wrist joint. The grate is attached to the robot at different lengths, to simulate different grasping locations. We demonstrate 20 movements per grasping location to train our approach. The data where recorded through tele-operation. In this experiment the state encodes the joint angle encoder value and the joint torque reading in the wrist. We present the recored torques from the sensor of the robot for all three demonstrated grasping locations in Fig. 8. By placing the grate on the same location as during the demonstration and reproducing the movement with our approach, we show that we observe the same torque profile. The force measurement is crucial in our experiment as it is used for applying the correct forces during the execution of the movement. When disabled, the robot either fails to lift the grate to the demonstrated location or it overshoots. The overshooting is due to gravity, as in that grasping location the center of the mass of the grate is moved over the axis of wrist rotation. The results are shown in Fig. 9. The reproduction distributions where created using twenty executions of the model-free ProMP controller per grasping location. Our approach can generalize to different grasping locations between the demonstrations. We generalized into four new locations and executed our controller. The robot reproduces the same joint distribution while compensating for the different dynamics, as shown in Fig.7.

V. Conclusion

In this paper, we presented a model-free approach for Probabilistic Movement Primitives (ProMP) that can be used for learning skills for physical interaction with the environment from demonstrations. In contrast to the original approach, the model-free ProMP approach does not require a known model of the system dynamics as the stochastic feedback controller is directly obtained from the estimated distribution over the trajectories, which includes the control signals. We showed that the model-free ProMP approach inherits many beneficial properties from the original ProMPs such as reproducing the variability in the demonstrations as well as using probabilistic operations such as conditioning for generalization to different via points. Our approach is different from directly encoding the actions, as generates the action through a model that depends on the state and the time. Hence, our approach can generalize well in the vicinity of the demonstrations. Our approach can be used in tasks where time is critical for the execution of the task, e.g. pushing a button at a specific movement, or grasping a moving object.

For learning physical interaction tasks, we showed that we can include sensory signals, for example the measure torques, in our distribution. By learning the correlations of this sensory signal, we can coordinate the controls needed for the physical interaction with the measured torques and forces. Such coordination is essential for the complex interaction tasks. In a preliminary study, we showed how the model-free ProMP approach can be applied to the iCub to apply forces to objects with unknown masses. In future work, we will investigate the use of model-free ProMPs for more complex scenarios.

References

[1] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[2] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Int. Symp. on Robotics Research.*, 2005.

[3] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*, 2008.

[4] J. Kober and J. Peters, "Learning motor primitives for robotics," in *Int. Conf. on Robotics and Automation (ICRA)*, 2009.

[5] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural Computation*, 2013.

[6] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[7] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, "Learning force control policies for compliant manipulation," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[8] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, 2014.

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

[10] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[11] A. Gams, M. Do, A. Ude, T. Asfour, and R. Dillmann, "On-line periodic movement and force-profile learning for adaptation to new surfaces," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2010.

[12] P. Kormushev, D. N. Nenchev, S. Calinon, and D. G. Caldwell, "Upper-body kinesthetic teaching of a free-standing humanoid robot," in *Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[13] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics and Automation Magazine*, Jun. 2010.

[14] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, 2011.

[15] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE Transactions on Haptics*, 2013.

[16] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[17] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: object-lifting case study with a humanoid," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2009.

[18] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *Int. Conf. on Robotics and Automation (ICRA)*, 2011.