

# Probabilistic Prioritization of Movement Primitives

Alexandros Paraschos<sup>1,2</sup>, Rudolf Lioutikov<sup>1</sup>, Jan Peters<sup>1,3</sup> and Gerhard Neumann<sup>4</sup>

**Abstract**—Movement prioritization is a common approach to combine controllers of different tasks for redundant robots, where each task is assigned a priority. The priorities of the tasks are often hand-tuned or the result of an optimization, but seldomly learned from data. This paper combines Bayesian task prioritization with probabilistic movement primitives to prioritize full motion sequences that are learned from demonstrations. Probabilistic movement primitives (ProMPs) can encode distributions of movements over full motion sequences and provide control laws to exactly follow these distributions. The probabilistic formulation allows for a natural application of Bayesian task prioritization. We extend the ProMP controllers with an additional feedback component that accounts inaccuracies in following the distribution and allows for a more robust prioritization of primitives. We demonstrate how the task priorities can be obtained from imitation learning and how different primitives can be combined to solve even unseen task-combinations. Due to the prioritization, our approach can efficiently learn a combination of tasks without requiring individual models per task combination. Further, our approach can adapt an existing primitive library by prioritizing additional controllers, for example, for implementing obstacle avoidance. Hence, the need of retraining the whole library is avoided in many cases. We evaluate our approach on reaching movements under constraints with redundant simulated planar robots and two physical robot platforms, the humanoid robot “iCub” and a KUKA LWR robot arm.

## I. INTRODUCTION

Complex robots with redundant degrees of freedom can, in principle, perform multiple tasks at the same time, e.g., reaching for an object with a humanoid robot while balancing, or reaching for an object with a robotic arm while avoiding an obstacle. However, simultaneously performing multiple tasks using the same degrees of freedom, requires combining the control signals from all tasks.

While many schemes for combining control signals have been developed, this paper focuses on approaches that resolve the control combination by prioritizing the tasks, i.e., approaches which assume that one task can be executed with priority over another task, even if the latter will not be performed sufficiently well.

In contrast to current approaches, where the priorities are set by experts, we propose learning the priorities from demonstrations. The Learning from Demonstrations (LfD) paradigm has been very successful in movement generation

\*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)

<sup>1</sup>Intelligent Autonomous Systems, TU Darmstadt, 64289 Darmstadt, Germany {paraschos, lioutikov}@ias.tu-darmstadt.de

<sup>2</sup>Data Lab, Volkswagen Group, 80805 Munich, Germany

<sup>3</sup>Robot Learning Group, Max Planck Institute for Intelligent Systems, Germany mail@jan-peters.net

<sup>4</sup>Lincoln Centre for Autonomous Systems (LCAS), University of Lincoln, UK gneumann@lincoln.ac.uk

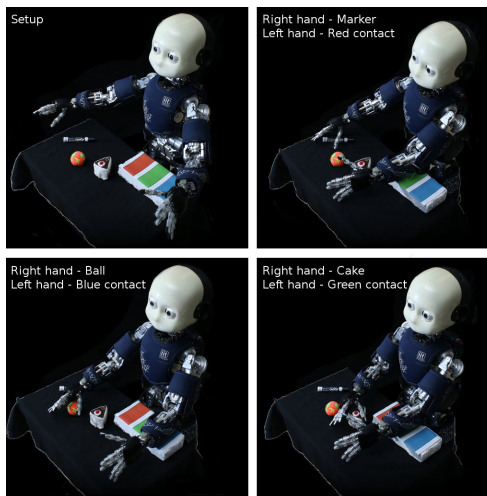


Fig. 1. The iCub robot performing a bi-manual reaching task. With the left end-effector, the robot initiates a supportive contact with the environment, while it performs a reaching task with the right end-effector. We illustrated our setup if the first picture. The robot stands on the floor and can choose three supporting contact locations, shown in blue, green, and red. With the right end-effector the robot reaches for one of three different objects, a marker, a ball, or a piece of cake. In the remaining pictures we present our results for learning and generalizing to new task combinations of reaching and contact support locations.

for complex robotic tasks. In LfD a solution is provided by demonstrations and, therefore, avoids using hand-coded controllers or the setting up a cost function for further planning or optimization. Yet, LfD has not been yet fully explored in prioritized control. LfD can introduce new properties to current prioritization approaches, such as adaptation to new situations and reproduction of unseen combinations of tasks, without retraining.

Further, in many robotic tasks, accurate reproduction for the whole duration of the task may not affect its performance. For example, a successful reproduction of a pick-and-place task, depends on highly accurate movements during picking and placing, but not for the rest of task’s execution. In classical approaches, where there is no notion of the task’s accuracy, combining tasks results in insufficient performance when the combination of both tasks is not physically possible. However, in our approach, we extract the time-varying task accuracy from demonstrations and we use it to modulate the task prioritization, where the accuracy implicitly defines the task’s priority. When a task has low accuracy, our approach allows for deviations from the reference trajectory and, therefore, enables tasks with higher accuracy to be executed at the same time without reaching the physical limitations of the system. Hence, our approach allows for a more efficient combination of tasks.

In this paper, we propose a novel data-driven framework

for learning prioritized task representations, i.e., learn the tasks and their relative priorities, from demonstrations. We combine Bayesian task prioritization [1] that allows the computation of the combined control signal from multiple tasks at different operational spaces with Probabilistic Movement Primitives (ProMPs) [2], [3] to learn prioritized complete motion sequences. MPs [2]–[6], are a powerful representation for encoding complex movements, much more flexible than using attractors or point-to-point movements, and enable adaptation of the learned movements without retraining. No primitive representation has so far taken advantage of introducing task priorities. ProMPs model the task and its desired accuracy from multiple demonstrations, provide a mechanism for adapting a learned task to novel situations, and an acceleration-space control law that follows the encoded task. The demonstrations can be acquired by several imitation learning techniques, including kinesthetic teach-in and tele-operation. We extend the Bayesian task prioritization [1] to provide a more general derivation for torque control and show that existing prioritization techniques are a special case of the Bayesian approach. We use the ProMP approach as it can naturally be combined with Bayesian task prioritization in a single, principled, probabilistic framework. We derive our approach based on ProMPs, however, other stochastic movement representations could be used analogously [5], [7].

We use our approach to learn multiple primitives for different operational spaces, e.g., the end-effector or the center of mass space. Each primitive solves a specific task in the corresponding space. We present how to adapt the task combination to new situations, e.g., reach a different via-point with an end-effector, that can be achieved without explicitly solving an inverse kinematics problem. Furthermore, we demonstrate how multiple primitives of different operational-spaces can now be seamlessly combined in order to achieve a new, unseen combination of tasks. Our prioritized LfD approach can be efficiently used without requiring re-learning the unseen combinations. As shown in our experiments, using prioritization also allows to adapt a library of primitives to changes in the environment, e.g., the introduction of an obstacle in the scene. We use simulators and real robot platforms to evaluate our approach.

## II. RELATED WORK

A common approach for combining different control signals is to prioritize the tasks, under the assumption that this prioritization is not allowed to be violated. We refer to such schemes as strict prioritization schemes. In these schemes, a higher priority task does not get disturbed by the control signals of the lower priority ones [8]–[15]. A lower priority task is always projected in the null-space of the high priority task. Although these approaches provide guarantees on the prioritization performance, strict prioritization approaches can get numerically unstable when the robot enters a singular kinematic configuration. Numerical regularization can be used, but it violates the null-space projection [16]. The introduction of regularization has the side-effect that a low

priority task can interfere with a higher priority task. We show how our approach can obtain similar regularizations from demonstrations.

For some tasks, such a prioritization scheme is natural, for example, a humanoid robot should not tip over and, therefore, the balancing controller should always have the highest priority. However, defining a strict priority can be problematic in general. For example, for reaching an object with one hand of a humanoid robot, while simultaneously reaching for a different object with the other hand, it is not clear these tasks can be prioritized. Both tasks could have the same importance, i.e. priority, or, the importance of each task could vary in time and depending on, e.g., the desired execution accuracy at that point in time. For such scenarios, the relative importance between the tasks can be easier to set. These problems are partially addressed in [17]–[19], where a “soft” prioritization scheme was introduced. In our approach, we step further and propose to learn the relative priorities from data and, therefore, minimize the amount of parameters that require tuning through expert knowledge.

“Soft” prioritization approaches do not assume a hierarchy of tasks a priori, but use the relative priorities between the tasks. In this scheme, every task contributes to the control signal. The degree of contribution depends on its relative priority. “Soft” prioritization approaches demonstrate promising results [17], [20], [21] to successfully perform multiple tasks due to the relaxation of the initial problem. Intuitively, “soft” prioritization schemes could be thought of as violating the hierarchy of priorities. They are often stated as multi-objective optimization problems [17], [20], [21]. Each task is formulated as a quadratic cost function and uses the relative priority as weight. The result of the optimization yields the controls that minimize the total cost and, therefore, allow lower priority tasks to perturb higher priority ones as long as the total cost is decreased.

Current prioritization approaches often assume a static prioritization or weighting scheme, where the importance of each task remains constant during the execution of the movement [10], [19]. However, modulating the importance of the tasks during the movement can be beneficial. First, tasks that are no longer desired to be executed can be faded-out and new tasks can be smoothly introduced, without torque jumps. Salini et al. [17] proposed to dynamically adjust the priorities for achieving movement sequencing and task transitions. More importantly, the modulation of the priorities can be related to the desired accuracy of the task. During the time-steps with low task-priority, the robot can focus on executing other tasks. Therefore, setting the relative priorities can be a simpler problem than specifying the strict task hierarchy, as the expert has to specify only the time points that require higher accuracy. Lober et al. [21] demonstrated that this approach increases the flexibility of the system and decreases “lock-ups” where a more important movement prohibits the execution of less important tasks, while requiring less expert knowledge. Modugno et al. [22] proposed the use of an optimization algorithm to find suitable relative priorities that further decreases expert knowledge.

### III. PROBABILISTIC PRIORITIZATION OF MOVEMENT PRIMITIVES

This paper presents a generic probabilistic framework for simultaneously combining multiple tasks. We assume that each task has a different level of accuracy and that the accuracy can change over the execution of the task. The task accuracy is associated with the respective importance for the task combination.

First, we encode the time-varying accuracy in an efficient representation and, importantly, obtain it from imitation data. Second, we develop our stochastic combination approach using the task accuracy as relative priorities. Furthermore, we show that current prioritization approaches can be derived within our probabilistic approach, when some uncertainty parameters are set to zero. Finally, we present an extension to the ProMPs controller that increases the tracking performance when prioritizing several primitives. We extend our approach to multiple operational-space controllers in Sec. III-C. In Sec. IV we show that our stochastic prioritizing scheme can be generalized to a wider class of controllers.

#### A. Encoding Task Accuracy from Demonstrations

Representing the desired task accuracy throughout the duration of the task is critical for relative prioritization schemes. A measure for the task accuracy is the task variance that is obtained over multiple executions of the task. Stochastic movement primitive representations can not only represent the task variance but also be trained from demonstration data. To this end, we use the Probabilistic Movement Primitives (ProMPs) approach [2], [3] as our representation.

ProMPs represent a single trajectory as a weighed linear combination of Gaussian basis functions  $\Phi_t$  and the respective weights  $w$ , i.e.,  $y_t = \Phi_t w$ , where  $y_t = [x, \dot{x}]^T$  represents the state of the task, i.e., positions and velocities, at time  $t$ . The task state  $y_t$  is a vector that contains the variables that define the state of the tasks, e.g., the joint or end-effector positions and velocities. Each task demonstration is used to estimate the weights  $w$  for that execution using a maximum likelihood approach [2]. From the set of estimated weights, ProMPs estimate a distribution over the weights, i.e.,

$$p(w) = \mathcal{N}(w | \mu_w, \Sigma_w), \quad (1)$$

which is assumed to be approximated well by a Gaussian, or a Gaussian mixture [23], [24]. Thus, ProMPs offer a compact representation of the trajectory distribution in task space, that is, the mean movement, the correlation between the task's variables, and their variance. With ProMPs, we can evaluate the distribution of the state  $p(y_t)$  at every time-step

$$p(y_t) = \int p(y_t | w) p(w) dw = \mathcal{N}(y_t | \mu_{y_t}, \Sigma_{y_t}) \quad (2)$$

in closed form. ProMPs also provide a stochastic linear feedback controller, which is also derived in closed form. The controller can follow the encoded task distribution exactly, i.e., it matches mean and variance of the distribution. In [2], [3], ProMPs are used to control the joints of the robot and, therefore, the controller outputs are joint torques. In

this paper, we generalize ProMPs to model and control in operational space, e.g., the robot's end-effector space. To do so, we adjust the ProMP controller's output to be the acceleration of task space variables. The stochastic controller is, therefore, given by

$$p(\ddot{x} | y_t) = \mathcal{N}(\ddot{x} | K_t y_t + k_t, \Sigma_{\ddot{x}}). \quad (3)$$

The mean of the controller is given by a linear feedback control law. The controller additionally contains the covariance of the task in the acceleration space, which plays an important part in our approach as it specifies the required accuracy of the control, see Sec. III-B. In summary, ProMPs are capable of representing and learning the task covariance  $\Sigma_y$ , and transforming it to the acceleration covariance  $\Sigma_{\ddot{x}}$ .

#### B. Probabilistic Combination of Tasks

We begin our derivation given two tasks, a joint-space task and an operational-space task. For each task, a stochastic controller is obtained from the corresponding ProMP that has been trained from demonstrations. Every output of each controller is normally distributed, i.e.,

$$p_1(\ddot{q}) \sim \mathcal{N}(\ddot{q} | \mu_{\ddot{q}}, \Sigma_{\ddot{q}}), \quad p_2(\ddot{x}) \sim \mathcal{N}(\ddot{x} | \mu_{\ddot{x}}, \Sigma_{\ddot{x}}). \quad (4)$$

The vector  $\ddot{q}$  denotes the joint acceleration for all of the joints of the robot and the vector  $\ddot{x}$  the operational-space acceleration. We drop the time-index for simplicity.

The operational-space controller and the joint-space controller can not be used simultaneously without accounting for the kinematics of the system. The system kinematics introduce a constraint between the operational and the joint space acceleration. The constraint is commonly defined in the velocity space by  $\dot{x} = J\dot{q}$ , where  $J$  denotes the Jacobian from a base-frame to the operational-space. Equivalently, by differentiation over time, we obtain the acceleration-space formulation  $\ddot{x} = J\ddot{q} + \dot{J}\dot{q}$  of the constraint, where  $\dot{J}$  denotes the time derivative of the Jacobian. Given the constraint in the acceleration-space, the operational-space controller depends on the current joint-acceleration  $\ddot{q}$ . The probability of the operational-space acceleration  $\ddot{x}$  given the joint acceleration  $\ddot{q}$  is defined as the conditional

$$p_{2|\ddot{q}}(\ddot{x} | \ddot{q}) \sim \mathcal{N}(\ddot{x} | J\ddot{q} + \dot{J}\dot{q}, \Sigma_{\ddot{x}}), \quad (5)$$

where the mean of the conditional distribution is given by the constraint and the variance is given by the desired task accuracy. We can now use the joint space ProMP as prior distribution and the desired task-space mapping  $p_{2|\ddot{q}}(\ddot{x} = \mu_{\ddot{x}} | \ddot{q})$  as likelihood to obtain the posterior distribution for the joint space controller using Bayes theorem, i.e.,

$$p_{1|\ddot{x}}(\ddot{q} | \ddot{x} = \mu_{\ddot{x}}) = \frac{p_{2|\ddot{q}}(\ddot{x} = \mu_{\ddot{x}} | \ddot{q}) p_1(\ddot{q})}{p_2(\ddot{x})} = \mathcal{N}(\ddot{q} | \mu, \Sigma).$$

The control law for the joint accelerations  $\ddot{q}$  is then obtained by computing the marginal distribution

$$p_{1|2}(\ddot{q}) = \int p_{1|\ddot{x}}(\ddot{q} | \ddot{x}) p_2(\ddot{x}) d\ddot{x} = \mathcal{N}(\ddot{q} | \mu'_{\ddot{q}}, \Sigma'_{\ddot{q}}). \quad (6)$$

The mean  $\mu'_{\dot{q}}$  and the covariance  $\Sigma'_{\dot{q}}$  are computed as

$$\mu'_{\dot{q}} = \mathbf{J}^\dagger (\mu_{\ddot{x}} - \dot{\mathbf{J}}\dot{q}) + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \mu_{\dot{q}} \quad (7)$$

$$\Sigma'_{\dot{q}} = (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \Sigma_{\dot{q}} + \mathbf{J}^\dagger \Sigma_{\ddot{x}} \mathbf{J}^{\dagger T} \quad (8)$$

where the  $\mathbf{J}^\dagger$  denotes the generalized inverse of the Jacobian

$$\mathbf{J}^\dagger = \Sigma_{\dot{q}} \mathbf{J}^T (\Sigma_{\ddot{x}} + \mathbf{J} \Sigma_{\dot{q}} \mathbf{J}^T)^{-1}. \quad (9)$$

In our approach, the joint space acceleration  $\ddot{q}$  and the task-space acceleration  $\ddot{x}$  are obtained from the stochastic feedback controller of the ProMPs. The variance of the task-space controller  $\Sigma_{\ddot{x}}$  is used as the regularization matrix and the variance of the joint-space controller  $\Sigma_{\dot{q}}$  as weighting. The regularization matrix  $\Sigma_{\ddot{x}}$  is full-rank.

### C. Extension to Multiple Tasks

Multiple operational-space controllers can be naturally integrated in our approach where each task  $i \in 1 \dots N$  can operate in a different operational space. In principle, it is sufficient to compute the posterior distribution over the joint acceleration  $\ddot{q}$ , given the accelerations of all task controllers  $\{\ddot{x}_i\}_{1 \dots N}$ , i.e.,  $p(\ddot{q} | \{\ddot{x}_i\}_{1 \dots N})$ , that can be computed recursively, or in a single step [1]. The single step solution does not relate to existing prioritization approaches.

For the recursive computation, we start with our prior distribution over the joint accelerations  $p_1(\ddot{q})$ . We condition it with the operational-space acceleration distribution  $p_N(\ddot{x}_N)$  of the highest priority task. The resulting posterior distribution  $p_{1|N}(\ddot{q} | \ddot{x}_N)$  is then used as a new prior distribution and is conditioned with  $p_{N-1}(\ddot{x}_{N-1})$ . We continue conditioning until we reach the task  $i = 1$ . During the computation of the new prior distribution, we can perform a numerical stability analysis of the matrix,  $\Sigma_{\ddot{x}} + \mathbf{J} \Sigma_{\dot{q}} \mathbf{J}^T$ , e.g., by computing the condition number of the matrix. If the inversion becomes numerically unstable, then the task  $i_o$  added at this step is incompatible to the already added tasks  $N \dots i_o - 1$ . The order of inference should be chosen by starting from the most important to the least important tasks, when physical limitations are reached, no more tasks would make sense to be added. Otherwise, the order of inference is not modify the resulting controller in our approach. Our recursive approach has similarities with the hierarchical prioritization approaches presented in [8], [9]. However, in our approach we use the regularized generalized inverse, as presented in Sec. III-B, where the tasks accuracies are obtained from imitation data, instead of treating each task with an infinite accuracy that can cause numerically unstable solutions.

### D. Robust Trajectory Distribution Tracking

We use the ProMP controller to get the desired accelerations in both, the task and joint spaces. The controller is based on [2], where the authors derive Eq. (3) by matching the change in the sufficient statistics of the system, i.e., change of mean and covariance, at the current time-step. Hence, it is assumed that  $\mu_t, \dot{\mu}_t, \Sigma_t$  and  $\dot{\Sigma}_t$  are known.

TABLE I

COMPARISON OF DIFFERENT PSEUDO INVERSES USED FOR OPERATIONS

$\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1}$	<b>Generalized inverse</b>
$\mathbf{J}^\dagger = \mathbf{M}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T)^{-1}$	<b>Weighted generalized inverse</b> , weighted with the inverse of the mass
$\mathbf{J}^\dagger = \Sigma_{\dot{q}} \mathbf{J}^T (\Sigma_{\ddot{x}} + \mathbf{J} \Sigma_{\dot{q}} \mathbf{J}^T)^{-1}$	<b>Bayesian inverse</b> , weighting and regu- larization are com- puted in closed form.

Due to the prioritization of multiple primitives, if deviations that were not present in the demonstrations occur and the system drift away from the demonstrated area. The controller computation presented in [2] generates gains that are not optimal for the drifted state distribution and these errors yield in an inaccurate tracking behavior, where the reproduction distribution does not match the demonstrated.

We propose to adjust the current belief of the mean state  $\mu_t$  and its derivative  $\dot{\mu}_t$ , according to the current observation of the state  $y_t$ . We adapt the mean belief  $\mu_t$  as a weighted average of the mean state obtained from demonstrations  $\mu_t^{\text{demo}}$  and the current state  $y_t$  as

$$\mu_t = \gamma \mu_t^{\text{demo}} + (1 - \gamma) y_t, \quad (10)$$

where  $\gamma$  is computed by a sigmoid activation based on the likelihood of the current state, i.e.,

$$\gamma(y_t) = (1 + \exp(-\log(p(y_t; \theta)) \beta^{-1} - \alpha))^{-1}, \quad (11)$$

where  $\alpha, \beta$  are open parameters. Additionally, we adapt the time derivative of the mean state  $\dot{\mu}_t$  with a feedback controller to converge to the demonstrated  $\dot{\mu}_t^{\text{demo}}$ , i.e.,

$$\dot{\mu}_t = \begin{bmatrix} \dot{\mu}_{\dot{q}_t} \\ \dot{\mu}_{\ddot{q}_t} \end{bmatrix} = \begin{bmatrix} K^{\text{SC}} (\mu_t^{\text{demo}} - \mu_t) + \mu_{\dot{q}_t}^{\text{demo}} \end{bmatrix}, \quad (12)$$

where  $K^{\text{SC}}$  are feedback gains. If the current state is inside the distribution,  $\gamma$  will be 1 and the correction term for the mean will not be activated. However, if the current state is outside the distribution, i.e., we have a small likelihood, it is an indication that inaccuracies in the controller computation accumulated, such that the distribution is not matched any more. In this case, the additional feedback controller is used to push the mean back to the desired mean, and, hence, increasing the tracking performance.

## IV. RELATION TO OPTIMIZATION-BASED PRIORITIZATION SCHEMES

Our approach can be generalized to a wider class of problems, where the constraints are linear to the joint acceleration  $\ddot{q}$ , i.e., can be formulated as  $\mathbf{A}\ddot{q} = \mathbf{b}$ , where the matrix  $\mathbf{A}$  and vector  $\mathbf{b}$  possibly depend on the current state of the robot  $q$  and  $\dot{q}$  at time  $t$ . The constraint imposed by the robot's mechanics can be re-formulated in the generalized form by setting  $\mathbf{A} = \mathbf{J}$  and  $\mathbf{b} = \ddot{x} - \dot{\mathbf{J}}\dot{q}$ .

We formulate an optimization problem that incorporates a soft version of this constraint while staying close to the prior

mean. The covariance matrices serve as L2 norm metrics for the objectives, i.e.,

$$\begin{aligned} \arg \min_{\dot{\mathbf{q}}} J = \arg \max_{\dot{\mathbf{q}}} & (\mathbf{A}\dot{\mathbf{q}} - \mathbf{b})^T \Sigma_{\ddot{\mathbf{x}}}^{-1} (\mathbf{A}\dot{\mathbf{q}} - \mathbf{b}) \\ & + (\dot{\mathbf{q}} - \boldsymbol{\mu}_{\dot{\mathbf{q}}})^T \Sigma_{\dot{\mathbf{q}}}^{-1} (\dot{\mathbf{q}} - \boldsymbol{\mu}_{\dot{\mathbf{q}}})^T. \end{aligned} \quad (13)$$

This formulation resembles the optimization framework presented in [8], but with  $\mathbf{A}\dot{\mathbf{q}} - \mathbf{b}$  imposed as soft-constraint and not as hard constraint. For  $\Sigma_{\ddot{\mathbf{x}}} = 0$ , we obtain a hard constraint and all the control laws in [8] can be recovered. The optimization view does not provide a direct way to update the joint covariance  $\Sigma_{\dot{\mathbf{q}}}$  if several tasks need to be prioritized, in contrast to the Bayesian approach.

#### A. Comparison to Strict Prioritization Approaches.

Our control law can be formulated for torques  $\mathbf{u}$  instead of desired accelerations  $\ddot{\mathbf{q}}$ . These derivations are given in the appendix. The mean  $\boldsymbol{\mu}_{\mathbf{u}}$  of the controls, given in Eq. (14), has a similar structure as well-known operational-space control laws [8]–[15]. It consists of a model-based component to compensate for the dynamics of the system, the desired acceleration in the operational-space—which, for example, can be the output of a feedback controller—and a projection component  $(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})$ .

The difference to the aforementioned approaches lays in the computation of the generalized inverse matrix of the Jacobian  $\mathbf{J}^\dagger$ . By applying a Bayesian approach, we obtain a generalized inverse matrix of the Jacobian which is both regularized and weighted, while strict prioritization methods use an un-regularized inverse.

All these related approaches can be derived by assuming that the operational-space variance  $\Sigma_{\ddot{\mathbf{x}}}$  is zero, i.e.  $\Sigma_{\ddot{\mathbf{x}}} = \lim_{\alpha \rightarrow 0} \alpha \mathbf{I}$  and, therefore, degrade our approach to a deterministic case. If the operational-space variance is zero, the matrix  $\mathbf{J}\mathbf{J}^\dagger = \mathbf{I}$  of the projection is a null-space projection, i.e. the lower priority tasks will not interfere with the higher priority tasks. Therefore, decreasing the variance of the operational-space controller  $\Sigma_{\ddot{\mathbf{x}}}$  can be interpreted as “hardening” the prioritization of the two controllers.

A consequence of not regularizing the generalized inverse is the numerical instability of the inversion at singular kinematic configurations. A regularization of the form  $\lambda \mathbf{I}$  is commonly used [16]. In our approach, this regularization has the physical interpretation of adapting the covariance of the operational space task. To our knowledge, the interpretation of this regularization has not been previously discussed.

By setting the joint-space covariance to  $\Sigma_{\dot{\mathbf{q}}} = \mathbf{I}$ , the pseudo inverse is un-regularized and unweighted and we can obtain control laws as in [8]–[13], [15]. Setting the joint-space covariance to  $\Sigma_{\dot{\mathbf{q}}} = \mathbf{M}^{-1}$ , we obtain controllers based on the Gauss principle of least constraint, and consistent to d’Alambert’s principle of virtual work [8], [14]. The different approaches for computing the generalized inverse are shown in Table I.

## V. EXPERIMENTAL EVALUATION

We evaluate our approach on redundant simulated and physical robots combining tasks learned by demonstrations.

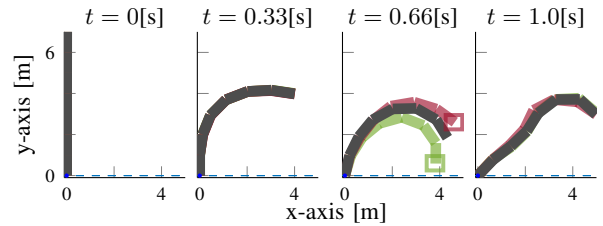


Fig. 2. A visualization of the 7-link planar robot trajectory for different time-steps. The dark configuration denotes the mean reproduction of the demonstrated primitive. In green and red we show the mean reproduction after conditioning at  $t = 0.66s$  to different via-points in end-effector space. The boxes show the targets of the end-effector.

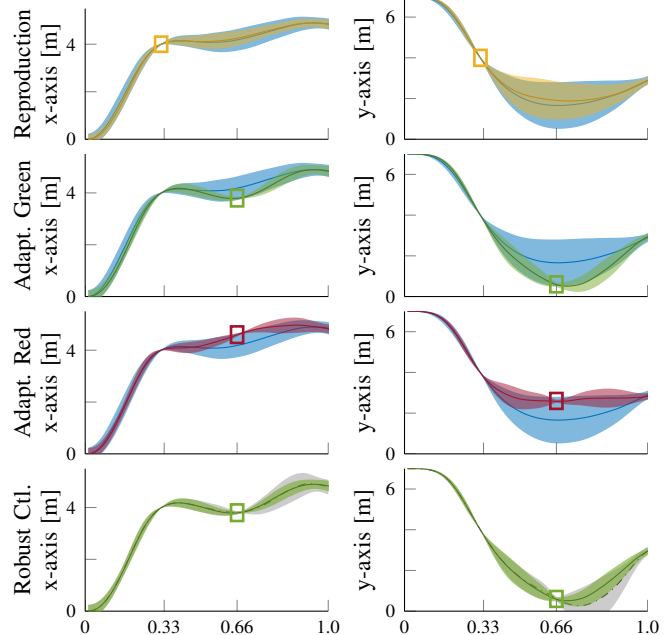


Fig. 3. We present the end-effector trajectory distribution of the 7-link planar robot evaluation. The demonstrated distribution is shown in blue. The shaded area represents two times the standard deviation. In the first row, the reproduced trajectory distribution, shown in yellow, follows accurately the demonstrations. The boxes illustrate the via-points present in the demonstrations. In the second and third rows, we present the reproduction distribution after adapting the primitive. The boxes illustrate the conditioning points. In the fourth row, we evaluate the performance of the proposed feedback controller. The reproduction distribution is shown in green and the reproduction of the original controller is shown in grey.

First, we demonstrate that our approach can be used for conditioning in operational space, an operation that was not feasible for the original ProMP approach. Second, we show that additional controllers can be smoothly integrated in our framework to implement additional constraints that were not present during training, e.g., for avoiding obstacles or keeping contact with another object. Third, we show how we learn a combination of tasks with considerably improved data efficiency and even generalize to unseen combinations.

#### A. Data-Driven Task-Space Adaptation

In this section, we adapt the learned task-space movement without explicitly solving the inverse kinematics problem. The adaptation, instead, is data-driven with the resulting trajectories staying close to the demonstrations. For the evaluation, we used a planar robot with seven degrees of freedom (DoFs) and optimal control to generate demonstrations in

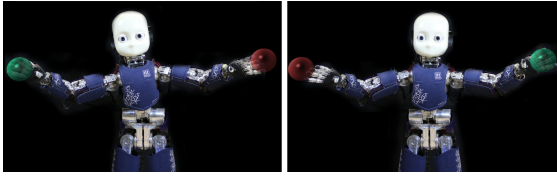


Fig. 4. The iCub robot performing a bi-manual reaching task while its “hip” stays fixed. The importance of the targets is time-varying, allowing the robot to perform all tasks.

joint-space. Optimal control allows for generating trajectories under the assumption that the cost function of the task is known. For a real application though, the cost function is typically not available.

We used ten demonstrations for training a joint-space ProMP and a task-space ProMP. The movement of the robot is visualized in Fig. 2. The demonstrations have different variability at different time-steps of the movement, e.g., at time step  $t = 0.33s$  the end-effector has low variability in both task-space dimensions. First, in Fig. 3 we show that our proposed controller can accurately track the demonstrated movement. Further, we adapt the task-space primitive by conditioning. Due to our prioritization scheme, no inverse kinematics algorithm is needed to find the respective joint configuration. Instead, the inverse kinematics problem is solved implicitly, in a data-driven way, using the prioritized controllers. We present our results in Fig. 2 and 3, where we adapt the learned primitive for two via-points. The reproduction can accurately pass through the via-points while it maintains the shape of the movement learned from the demonstrations. We compare the performance of the proposed robust feedback controller to the controller proposed in [2] in Fig. 3, where the latter shows an improved tracking performance.

### B. Incorporation of External Control Laws

Expert-knowledge can be incorporated in our approach to adapt the learned primitives. We demonstrate our approach on a planar robot with two end-effectors that has three links for the torso and five links for each arm. Each link is one meter long. The robot tracks a “hip” movement that stays at a constant height of  $2.5m$ . An expert designed two feedback controllers with high gains and low variance  $\Sigma_u = 10^3$  that attract end-effectors at  $\{2, 6\}(m)$  and  $\{-2, 6\}(m)$ , respectively. The resulting movement is shown in Fig. 5, where the robot performs successfully all of the three tasks; it reproduces accurately the hip movement staying at the desired height and places its end-effectors at the desired locations set by the expert.

Similarly, we evaluated our approach on the “iCub” robot. We defined a target for each hand of the robot to be reached with high accuracy at different time points. The hip of the robot should remain fixed for maintaining balance. The robot can successfully perform all tasks, as shown in Fig. 4.

### C. Increased data efficiency

We demonstrate the increased learning efficiency of our approach in combining tasks of different end-effectors, over the traditional approach of learning all task combinations

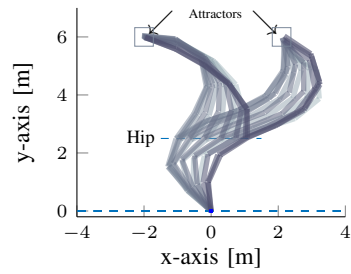


Fig. 5. The planar robot performing a bi-manual reaching task while moving its “hip”, by combining three tasks, the two end-effector tasks and the hip task. The robot accurately stays at the desired targets with its end-effectors during the movement while the “hip” tracks a trajectory to remain at constant height.

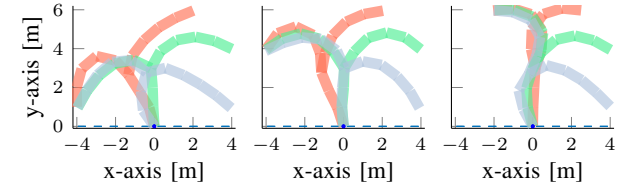


Fig. 6. A visualization of the two end-effector robot. We present the final configuration,  $t = t_{end}$ , of the robot in task space for all nine different tasks.

independently. For our experiments we used the planar robot with two end-effectors and thirteen DoF. The two concurrent end-effector tasks are not independent, as they control the common joints of the torso.

First, we generated demonstrations where each end-effector reaches one out of three end-points at  $t_{end} = 1$ . The end-point can either be set at  $\{\pm 4, 1\}$ ,  $\{\pm 4, 4\}$ , or  $\{\pm 2, 6\}$ . An illustration of the configuration of the robot at these points is shown in Fig. 6. The combination of all three tasks of the two end-effectors yields nine different task combinations. For each combination, we generate a set of noisy demonstrations. As a baseline, we train nine individual primitives, one for each combination of tasks. In contrast, our approach can use all available demonstrations per task of a single end-effector as it can learn the end-effector tasks independently resulting in three times as much training data per task. Therefore, our approach considerably outperforms the baseline as the distributions can be estimated with higher accuracy, as shown in Fig. 7. We evaluate the average performance of both approaches which was specified as the negative square deviation from the true desired task-space position at the end of the movement.

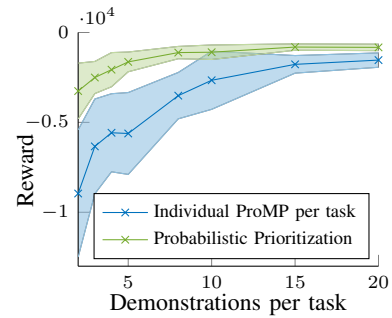


Fig. 7. Comparison between the prioritization of MPs and learning each task combination independently, used as a baseline. We vary the number of demonstrations used per task combination. Using prioritization, we can learn the tasks for each end-effector independently, and, therefore, use more training data per task. The baseline can only learn each combination of the end-effector tasks individually. The plot shows the average negative square deviation from the true end-effector position averaged over ten trials.

TABLE II  
INITIATING CONTACTS DURING REACHING EVALUATION

	Left Task Err. (cm)	Right Task Err. (cm)
Blue — Marker	$2.38 \pm 0.91$	$3.09 \pm 1.22$
Blue — Ball	$2.34 \pm 0.96$	$3.18 \pm 1.10$
Blue — Cake	$2.05 \pm 0.71$	$3.56 \pm 1.45$
Green — Marker	$2.21 \pm 0.64$	$1.70 \pm 0.81$
Green — Ball	$2.47 \pm 0.89$	$2.28 \pm 1.26$
Green — Cake	$2.97 \pm 0.84$	$3.85 \pm 1.02$
Red — Marker	$3.67 \pm 0.76$	$2.89 \pm 1.66$
Red — Ball	$2.82 \pm 0.75$	$2.43 \pm 1.13$
Red — Cake	$3.31 \pm 1.26$	$4.23 \pm 1.62$

#### D. Initiating Contacts during Reaching

We performed a bi-manual experiment using the “iCub” to reach objects while improving its stability by partially supporting its weight on a table. The experiment is difficult to demonstrate using kinesthetic teaching, as the teacher would have to simultaneously move both arms of the robot and keep track of the torso configuration. However, using the decoupling properties of our approach the task can be demonstrated with ease, a single arm at a time. Additionally, the decoupling approach utilizes improved data-efficiency, as it was shown in Sec. V-C, and allows to generalize in novel movement combinations.

Reaching objects that require the robot to bend the torso can move the center of gravity of the robot out of the support polygon defined by the feet, and, as a result, the robot will lose its balance. The task of the robot is to perform a reaching movement while it initiates a contact to stabilize itself. With its right arm reaches for three different objects, as shown in Fig. 1. Concurrently, with the left arm, initiates a contact with the table that increases its stability. The location of the contact varies over three positions. We provided ten demonstrations for reaching each object and for each supportive contact location. The robot was capable of reproducing the movements using the prioritized movement primitives and generalizing to task combinations not present in the demonstrations. The reaching performance is shown in Table II.

#### E. Adaptation of a Movement Primitive Library

We demonstrate how to adapt a movement primitive library to changes of the environment, e.g., an obstacle that was not present during training. In our setup, Fig. 8, we used a seven DoF humanoid arm mounted on a fixed base, with a hand as its end-effector. The hand was not controlled during the experiments. We trained the robot with three distinct movements using kinesthetic teaching. First, approach a bottle for grasping, second, drop a peg into a bowl, and, third, push a button. Each primitive can generalize into a  $25\text{cm} \times 25\text{cm}$  area. We provided ten demonstrations for each primitive. After training the primitives, we introduce an obstacle, the cat, into the environment. The robot collides with the cat during the execution of the primitives. To adapt the library, we demonstrated an additional primitive that

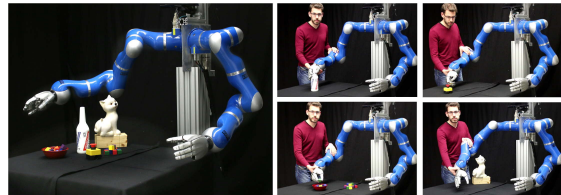


Fig. 8. The setup used for adapting a primitive library due to environmental changes. The robot learns how to grasp a bottle, place a peg into a bowl, and push a button with kinematic teaching. The cat is then placed in a position that collides with the robot during reproduction.

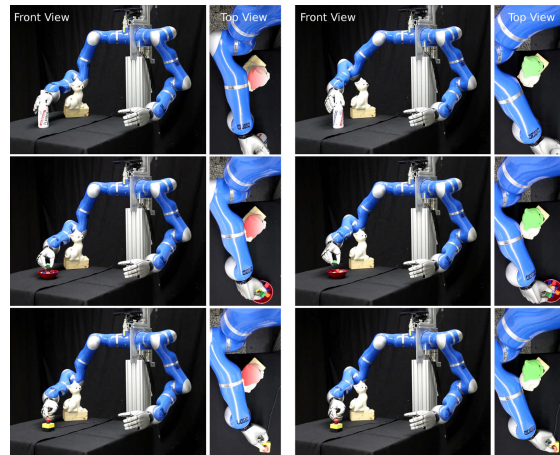


Fig. 9. In the left, each row shows the reproduction of the respective primitive after training and the introduction of the obstacle without adaptation. The robot collides with the head of the cat. In the right, we present the reproduction after adaptation where the robot avoids collisions.

avoids collisions with the obstacle, but without any of the three objects present in the environment during the demonstrations. While the robot does not perceive the obstacle, the information of avoiding it is encoded in the new primitive. By combining the new primitive with each of the primitives in the library, we avoid collisions with the obstacle, as shown in Fig. 9. The success rate of the experiment is presented in Table III, averaging over ten reproductions per case. Finally, we evaluated classical prioritization approaches on our system. We modelled the mean trajectory using ProMPs, set  $\Sigma_{\dot{q}} = \mathbf{I}$ , and  $\Sigma_{\ddot{x}} = \lambda \mathbf{I}$ . The value of  $\lambda$  was optimized to reduce the Cartesian error at the end of the movement. The tasks could not be effectively combined using standard methods [8]–[16], as shown in Table III.

TABLE III  
ADAPTATION OF A MP LIBRARY — TASK EVALUATION

	Bottle	Peg	Button
Obstacle Avoidance Rate			
Before Adaptation	0.0	0.1	0.3
After Adaptation	0.9	1.0	1.0
Std. Approaches	1.0	1.0	1.0
Avg. End-Effector Error (cm)			
After Adaptation	$0.48 \pm 0.23$	$0.67 \pm 0.23$	$0.45 \pm 0.51$
Std. Approaches	$12.83 \pm 4.57$	$3.21 \pm 1.45$	$2.12 \pm 1.53$

## VI. CONCLUSION

In this paper, we presented a novel approach for movement prioritization based on the combination of Bayesian task prioritization and the Probabilistic Movement Primitives. While prioritization is a well established concept in control, it has not been explored in the context of learning movements from demonstrations. We have shown that combining prioritization with learning approaches yields in a powerful representation that can be used to solve a combination of tasks with different end-effectors. Our approach is data-driven, i.e., it can solely be trained from demonstrations and minimizes expert knowledge. It avoids the problem of specifying a cost function for the task at hand, which is still an open problem. We demonstrated that our approach can be used to adapt task-space movements without solving an inverse kinematics problem and, importantly, staying close to the demonstrated data. Further, we propose an extension to the ProMP controller that can handle deviations that occur from the demonstrated movements due to the prioritization.

A key contribution of our approach is the ability to combine tasks of different end-effectors in a principled and data-efficient way. Our approach can generalize to task combinations that were not present in the demonstrations and requires significantly less training data to achieve the same level of performance. Our approach can be used to adapt a library of primitives without extensive retraining.

Given that our approach is data-driven, it heavily relies on quality demonstrations. If the task is too difficult to be demonstrated or if non-informative demonstrations are provided, our approach will match the provided data and not the intention of the user.

In future work, we will expand the evaluations of our approach on more complex real-world scenarios. We consider multiple task execution with physical robot interaction under the present of contacts as interesting research direction.

## APPENDIX

### INCLUDING THE DYNAMICS OF THE SYSTEM

The stochastic controller on the joint acceleration given in Eq. (6) can be used to control a physical system, i.e., by torque control, using the rigid-body dynamics model [25],  $\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q})$ , where  $\mathbf{M}(\mathbf{q})$  denotes the inertia matrix,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  denotes Coriolis and centripetal forces, and  $\mathbf{G}(\mathbf{q})$  forces due to gravity. Using the rigid-body dynamics model, we reformulate our controller to operate in the joint torque space, i.e.  $p_{1|2}(\mathbf{u}) = \mathcal{N}(\mathbf{u} | \boldsymbol{\mu}'_{\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{u}})$ . The mean  $\boldsymbol{\mu}'_{\mathbf{u}}$  of this controller is given by

$$\boldsymbol{\mu}'_{\mathbf{u}} = \mathbf{M} \left( \mathbf{J}^\dagger \left( \boldsymbol{\mu}_{\ddot{\mathbf{x}}} - \dot{\mathbf{J}}\dot{\mathbf{q}} \right) + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \boldsymbol{\mu}_{\ddot{\mathbf{q}}} \right) + \mathbf{C} + \mathbf{G},$$

where we used  $\boldsymbol{\mu}_{\ddot{\mathbf{q}}} = \mathbf{M}^{-1}(\boldsymbol{\mu}_{\mathbf{u}} - \mathbf{C} - \mathbf{G})$ . Furthermore, decoupling of the kinematics and the dynamics can be obtained by setting  $\hat{\boldsymbol{\mu}}_{\mathbf{u}} = \boldsymbol{\mu}_{\mathbf{u}} + \mathbf{C} + \mathbf{G}$  and using it in place of  $\boldsymbol{\mu}_{\mathbf{u}}$ . In this case, the mean becomes

$$\boldsymbol{\mu}'_{\mathbf{u}} = \mathbf{M} \mathbf{J}^\dagger (\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{M}(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})(\mathbf{M}^{-1}\boldsymbol{\mu}_{\mathbf{u}}) + \mathbf{C} + \mathbf{G} \quad (14)$$

that results in the resolved-acceleration controller [26], [27].

## REFERENCES

- [1] M. Toussaint and C. Goerick, "A bayesian view on motor control and planning," in *From Motor Learning to Interaction Learning in Robots*, 2010.
- [2] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Neural Inf. Proc. Sys.*, 2013.
- [3] A. Paraschos, G. Neumann, and J. Peters, "A probabilistic approach to robot trajectory generation," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2013.
- [4] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.
- [5] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Int. Conf. on Humanoid Robots*, 2012.
- [6] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *Transactions on Robotics*, 2011.
- [7] L. Rozo, S. Calinon, D. Caldwell, P. Jiménez, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *AAAI Conference on Artificial Intelligence*, 2013.
- [8] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal, "A unifying framework for robot control with redundant DOFs," *Autonomous Robots*, 2007.
- [9] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Journal of Robotics and Automation*, 1987.
- [10] O. Khatib, L. Sentis, J. Park, and J. Warren, "Whole-body dynamic behavior and control of human-like robots," *International Journal of Humanoid Robotics*, 2004.
- [11] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-Priority based redundancy control of robot manipulators," *International Journal of Robotics Research (IJRR)*, 1987.
- [12] L. Sentis and O. Khatib, "Synthesis of Whole-Body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, 2005.
- [13] J. Park, W.-K. Chung, and Y. Youm, "Characterization of instability of dynamic control for kinematically redundant manipulators," in *International Conference on Robotics and Automation (ICRA)*, 2002.
- [14] H. Bruyninckx and O. Khatib, "Gauss' principle and the dynamics of redundant and constrained manipulators," in *International Conference on Robotics and Automation (ICRA)*, 2000.
- [15] J. Luh, M. Walker, and R. Paul, "Resolved-acceleration control of mechanical manipulators," *Transactions on Automatic Control*, 1980.
- [16] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 1998.
- [17] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: a focus on sequencing and tasks transitions," in *International Conference on Robotics and Automation (ICRA)*, 2011.
- [18] J. Salini, S. Barthélemy, P. Bidaud, and V. Padois, "Whole-Body motion synthesis with LQP-Based controller – application to icub," in *Modeling, Simulation and Optimization of Bipedal Walking*, 2013.
- [19] W. Decre, R. Smits, H. Bruyninckx, and J. D. Schutter, "Extending iTaSC to support inequality constraints and non-instantaneous task specification," in *Int. Conf. on Robotics and Automation (ICRA)*, 2009.
- [20] R. Lober, V. Padois, and O. Sigaud, "Multiple task optimization using dynamical movement primitives for whole-body reactive control," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2014.
- [21] —, "Variance modulated task prioritization in Whole-Body control," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.
- [22] V. Modugno, G. Neumann, E. Rueckert, G. Oriolo, J. Peters, and S. Ivaldi, "Learning soft task priorities for control of redundant robots," in *Int. Conf. on Robotics and Automation (ICRA)*, 2016.
- [23] M. Ewerton, G. Neumann, R. Lioutikov, H. Ben Amor, J. Peters, and G. Maeda, "Learning multiple collaborative tasks with a mixture of interaction primitives," in *ICRA*, 2015.
- [24] E. Rueckert, J. Mundo, A. Paraschos, J. Peters, and G. Neumann, "Extracting Low-Dimensional control variables for movement primitives," in *Int. Conf. on Robotics and Automation*, 2015.
- [25] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.
- [26] T. Yoshikawa, *Foundations of robotics: analysis and control*. MIT Press, 1990.
- [27] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," in *Int. Conf. on Robotics and Automation*, 1988.