Hybrid Probabilistic Trajectory Optimization Using Null-Space Exploration

Yanlong Huang, João Silvério, Leonel Rozo, and Darwin G. Caldwell

***Abstract*— In the context of learning from demonstration, human examples are usually imitated in either Cartesian or joint space. However, this treatment might result in undesired movement trajectories in either space. This is particularly important for motion skills such as striking, which typically imposes motion constraints in both spaces. In order to address this issue, we consider a probabilistic formulation of dynamic movement primitives, and apply it to adapt trajectories in Cartesian and joint spaces simultaneously. The probabilistic treatment allows the robot to capture the variability of multiple demonstrations and facilitates the mixture of trajectory constraints from both spaces. In addition to this proposed hybrid space learning, the robot often needs to consider additional constraints such as motion smoothness and joint limits. On the basis of Jacobian-based inverse kinematics, we propose to exploit robot null-space so as to unify trajectory constraints from Cartesian and joint spaces while satisfying additional constraints. Evaluations of hand-shaking and striking tasks carried out with a humanoid robot demonstrate the applicability of our approach.**

I. Introduction

Imitation learning, also known as Learning from Demonstration, is an approach to easily teach robot various skills [1]. Depending on the specific task requirements and human interpretation of tasks, imitation learning can be carried out in either joint [2] or Cartesian space [3]. While imitation learning in a single space has achieved reliable performance, the simultaneous learning of skills in both spaces (which we refer to as *hybrid space learning*) has not been sufficiently investigated yet. In order to illustrate the importance of this hybrid approach, let us consider a robot table tennis task [4], where a racket is attached to the end-effector of an anthropomorphic robot arm. The preliminary goal is to control the racket (being held by the robot end-effector) so as to return the ball towards the human opponent side. In this case, imitation learning in the robot Cartesian space (i.e., learning of racket trajectory) is crucial. However, in order to mimic the human striking movement in joint space, we still need to take the robot posture into account, which is explicitly determined by the joint trajectories and their correlations. Besides the aforementioned scenario, other applications such as robot grasping task [5] and dual-arm manipulation [6] also demonstrated the advantages of hybrid space learning.

In this paper, we address the imitation learning in both Cartesian and joint spaces, where the probabilistic treatment of dynamic movement primitive (DMP) proposed in [3], [7] is exploited. The probabilistic DMP essentially integrates the DMP stability property and the probabilistic encoding of Gaussian mixture model (GMM). In contrast to the classical DMP [8] that models the forcing term as an explicit function of the phase variable using basis functions, the probabilistic DMP models the joint probability distribution of the phase variable and forcing term using GMM, and subsequently retrieves a desired forcing term using Gaussian mixture regression (GMR). Therefore, this approach encapsulates the consistent features underlying multiple demonstrations as well as the correlations between variables, which can be used to coordinate high-dimensional movement trajectories.

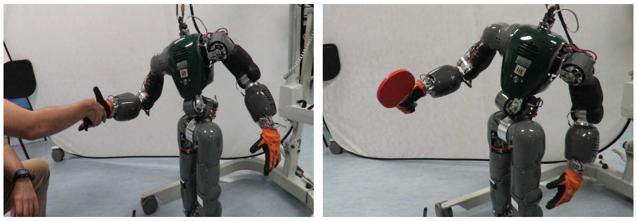


Fig. 1. Illustrations of hand-shaking (*left* graph) and striking tasks (*right* graph), where movements in both Cartesian and joint spaces are relevant.

On the basis of the probabilistic DMP, we propose to exploit robot null-space so as to unify both Cartesian and joint constraints, since the exploration of null-space allows for a flexible joint trajectory in a redundant robotic system. More specifically, we introduce a covariance-weighted measure to guide the null-space exploration so as to provide the robot with trajectories that closely resemble human demonstrated trajectories in both Cartesian and joint spaces. Moreover, the proposed hybrid space learning framework is extended to include additional joint constraints (e.g., movement smoothness and joint limits) in order to facilitate a robust robot trajectory execution.

This paper is organized as follows. We first review the probabilistic DMP in Section II. Subsequently, we propose the hybrid trajectory optimization using null-space exploration in Section III, and illustrate the performance of our approach in Section IV. Section V covers related work and, finally, we summarize our results and discuss possible extensions in Section VI.

II. A Probabilistic Treatment of Dynamical Movement Primitives

When learning a skill from human demonstrations, robot trajectories in Cartesian and joint spaces can be recorded through kinesthetic teaching, which are subsequently used to train a DMP. The trained DMP allows for adapting

All authors are with Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genoa, Italy, firstname.lastname@iit.it

This work was supported by the Italian Minister of Defense.

trajectories in both spaces towards new goals while providing a stable reproduction of the task.

Formally, let us consider a task exclusively learned in Cartesian space as an example, and denote the Cartesian position of the robot end-effector as $\boldsymbol{\xi}$. Formally, DMP is described as (see [8] for details)

$$\begin{aligned}\tau\dot{s} &= -\alpha s\\ \tau^2\ddot{\boldsymbol{\xi}} &= \mathbf{K}^p(\mathbf{g}_p - \boldsymbol{\xi}) - \tau\mathbf{K}^v\dot{\boldsymbol{\xi}} + s\mathbf{f}^p(s)\\ f_k^p(s) &= \frac{\sum_{i=1}^N \phi_i(s)w_{k,i}}{\sum_{i=1}^N \phi_i(s)}\end{aligned}, \tag{1}$$

where $\tau > 0$ and $\alpha > 0$ respectively represent the movement duration and a scalar, s is the phase variable driving the forcing term $\mathbf{f}^p(s)$.[1] Moreover, $f_k^p(s)$ denotes the k-th element of $\mathbf{f}^p(s)$, $\phi_i(s) = e^{-h_i(s-c_i)^2}$ are the basis functions with parameters $h_i > 0$ and $c_i \in [0,1]$, $w_{k,i}$ represents the weighting coefficient, and N denotes the number of basis functions. Finally, $\mathbf{g}_p$ represents the trajectory goal, $\mathbf{K}^p$ and $\mathbf{K}^v$ can be viewed as stiffness and damping diagonal matrices, while $\dot{\boldsymbol{\xi}}$ and $\ddot{\boldsymbol{\xi}}$ denote the Cartesian velocity and acceleration, respectively.

DMP essentially models the relationship between $\ddot{\boldsymbol{\xi}}_t$ and $\{\boldsymbol{\xi}_t, \dot{\boldsymbol{\xi}}_t\}$ using a spring-damper system dynamics while the phase variable s and the forcing term $\mathbf{f}(s)$ are used to modulate the movement duration and trajectory shape, respectively. Note that we slightly modify the original DMP in [8], so that we can introduce the probabilistic treatment of DMP conveniently.

Instead of explicitly modeling the forcing term $\mathbf{f}^p$ as a function of the phase variable s, we can model the joint probability distribution of $\{s, \mathbf{f}^p\}^\top$ (see [3], [7]), which alleviates the need for basis functions $\phi_i(s)$. Formally, let us assume that M demonstrations of time-length N are recorded and denoted by $\{\{t_{n,m}, \boldsymbol{\xi}_{n,m}\}_{n=1}^N\}_{m=1}^M$. By using the collected dataset and applying (1), a new dataset $\{s_k, \mathbf{f}_k^p\}_{k=1}^{M\times N}$ is obtained, which is subsequently used to train a GMM that models the joint probability distribution $\mathcal{P}(s, \mathbf{f}^p)$, yielding $\begin{bmatrix} s\\ \mathbf{f}^p\end{bmatrix} \sim \sum_{i=1}^K \pi_i^p \mathcal{N}(\boldsymbol{\mu}_i^p, \boldsymbol{\Sigma}_i^p)$ with prior probabilities π_i^p, means $\boldsymbol{\mu}_i^p$ and covariances $\boldsymbol{\Sigma}_i^p$.

Each Gaussian component $\{\boldsymbol{\mu}_i^p, \boldsymbol{\Sigma}_i^p\}$ can be re-written into a block-decomposition form, i.e., $\boldsymbol{\mu}_i^p = \begin{bmatrix}\boldsymbol{\mu}_{s,i}^p\\ \boldsymbol{\mu}_{f,i}^p\end{bmatrix}$ and $\boldsymbol{\Sigma}_i^p = \begin{bmatrix}\boldsymbol{\Sigma}_{ss,i}^p & \boldsymbol{\Sigma}_{sf,i}^p\\ \boldsymbol{\Sigma}_{fs,i}^p & \boldsymbol{\Sigma}_{ff,i}^p\end{bmatrix}$. For a new datapoint s, its corresponding conditional output is computed by GMR as follows (see [9])

$$\mathcal{P}(\mathbf{f}^p|s) = \sum_{i=1}^K h_i^p(s)\mathcal{N}(\bar{\boldsymbol{\mu}}_i^p(s), \bar{\boldsymbol{\Sigma}}_i^p), \tag{2}$$

with

$$h_i^p(s) = \frac{\pi_i^p \mathcal{N}(s|\boldsymbol{\mu}_{s,i}^p, \boldsymbol{\Sigma}_{ss,i}^p)}{\sum_{k=1}^K \pi_k^p \mathcal{N}(s|\boldsymbol{\mu}_{s,k}^p, \boldsymbol{\Sigma}_{ss,k}^p)} \tag{3}$$

$$\bar{\boldsymbol{\mu}}_i^p(s) = \boldsymbol{\mu}_{f,i}^p + \boldsymbol{\Sigma}_{fs,i}^p(\boldsymbol{\Sigma}_{ss,i}^p)^{-1}(s - \boldsymbol{\mu}_{s,i}^p) \tag{4}$$

$$\bar{\boldsymbol{\Sigma}}_i^p = \boldsymbol{\Sigma}_{ff,i}^p - \boldsymbol{\Sigma}_{fs,i}^p(\boldsymbol{\Sigma}_{ss,i}^p)^{-1}\boldsymbol{\Sigma}_{sf,i}^p. \tag{5}$$

By following the same approach we can fit M demonstrated $\mathcal{D}$-dimensional joint trajectories $\{\{t_{n,m}, \mathbf{q}_{n,m}\}_{n=1}^N\}_{m=1}^M$, where $\mathbf{q} \in \mathbb{R}^{\mathcal{D}}$ represents the robot joint position. In this case we model $\mathcal{P}(s, \mathbf{f}^q)$ and retrieve the conditional probability $\mathcal{P}(\mathbf{f}^q|s)$ using GMR. It is worth pointing out that the prediction in (2) can be approximated by a single Gaussian distribution [9], which facilitates the combination of trajectory constraints coming from both Cartesian and joint spaces, as explained in the next section.

III. Hybrid Probabilistic Trajectory Optimization

Given trained probabilistic DMPs with joint probability distributions $\mathcal{P}(s, \mathbf{f}^p)$ and $\mathcal{P}(s, \mathbf{f}^q)$, we need to adapt them to new goals in order to mimic human skills in Cartesian and joint spaces simultaneously. The problem is that, for every new Cartesian goal that was not observed during the demonstrations, the robot does not know the joint goal that best resembles the robot postures as demonstrated.

Let us picture a reaching task as an example. The trajectory adaptation in the Cartesian space is straightforward, where we only need to substitute the new position of the object $\mathbf{g}_p^*$ (i.e., the target in Cartesian space) into (1). However, the adaptation in joint space is less intuitive since the desired joint state at the end of the task (i.e., the target in joint space) is unknown beforehand. More specifically, when the robot is redundant and thus its inverse kinematics has infinite solutions, it is non-trivial to determine an appropriate joint target $\mathbf{g}_q^*$ that is consistent with $\mathbf{g}_p^*$.

In order to determine the desired joint target $\mathbf{g}_q^*$, we propose to explore the null-space of the robot Jacobian, in order to make full use of the robot redundancy (Section III-A). By exploiting the null-space, we can unify the adapted Cartesian and joint trajectories directly, as discussed in Section III-B. Note that we might encounter additional constraints such as motion smoothness and joint limits, thus we formulate the optimization in the null-space as a reinforcement learning problem so as to address the hybrid learning and additional constraints (Section III-C). An illustration of the proposed hybrid space learning is shown in Fig. 2.

A. Trajectory Adaptation in Cartesian and Joint Spaces

Let us first consider the trajectory adaptation in Cartesian space, and assume that the forcing term $\mathbf{f}_t^p \sim \mathcal{N}(\boldsymbol{\mu}_{f,t}^p, \boldsymbol{\Sigma}_{f,t}^p)$ at time t has been retrieved.[2] By substituting the new target $\mathbf{g}_p^*$ and the forcing term into the DMP model (1), we can estimate the corresponding acceleration $\ddot{\boldsymbol{\xi}}_t$, as well as $\dot{\boldsymbol{\xi}}_t$ and $\boldsymbol{\xi}_t$ by numerical integration.

With the new Cartesian target $\mathbf{g}_p^*$, we employ an inverse kinematics controller to estimate the joint trajectories that correspond to the desired trajectory in Cartesian space, i.e.,

$$\mathbf{q}_t = \mathbf{q}_{t-1} + \mathbf{J}^\dagger(\mathbf{q}_{t-1})(\boldsymbol{\xi}_t - \boldsymbol{\xi}_{t-1}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\mathbf{N}(\boldsymbol{\theta})\delta_t, \tag{6}$$

[1] The superscript 'p' represents Cartesian space component.

[2] Time t should be transformed into s before retrieval using GMR

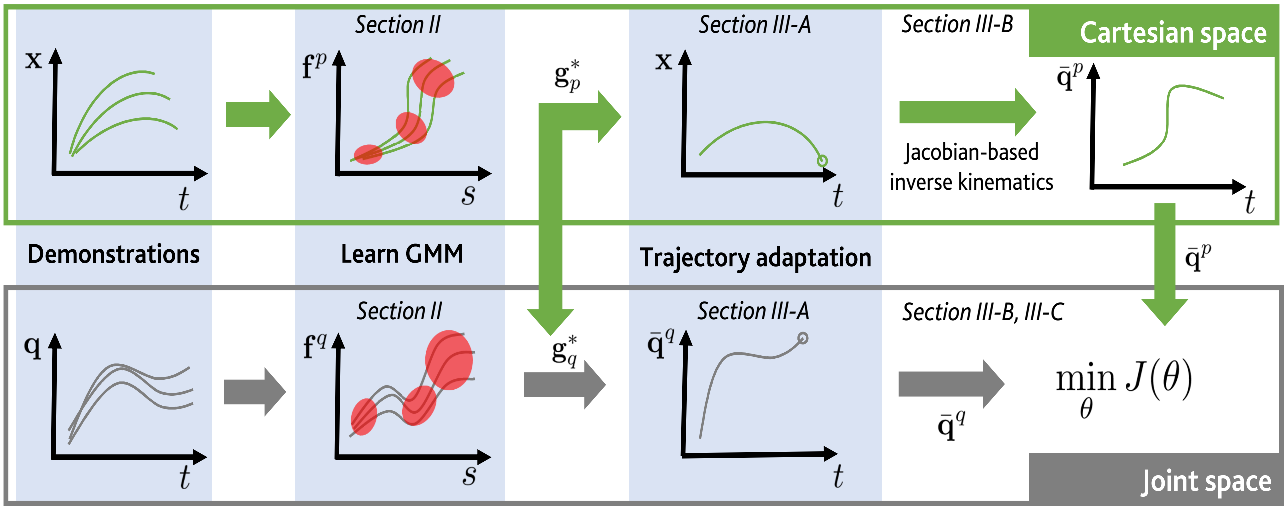


Fig. 2. Illustration of the hybrid space learning. Human demonstrations are used to extract datasets $\{s_k, \mathbf{f}_k^p\}_{k=1}^{M\times N}$ and $\{s_k, \mathbf{f}_k^q\}_{k=1}^{M\times N}$ in Cartesian and joint spaces, respectively. GMM is then employed to fit these datasets. After, given a new Cartesian target, we first estimate the corresponding joint target and then generate adapted trajectories in both spaces using (1). The adapted Cartesian trajectory is transformed into joint space by using the Jacobian-based inverse kinematics. Finally, the null-space parameter $\boldsymbol{\theta}$ is optimized to unify trajectory constraints from both spaces as well as to fulfill additional constraints.

where $\mathbf{J}^\dagger = \mathbf{J}^\top(\mathbf{J}\mathbf{J}^\top)^{-1}$ is the Moore-Penrose pseudo-inverse of $\mathbf{J}$, $\mathbf{I}$ represents the identity matrix of proper dimensionality, $\mathbf{N}(\boldsymbol{\theta})$ corresponds to a joint space movement in the null-space that is parameterized by the null-space parameter $\boldsymbol{\theta}$, and $\delta_t > 0$ denotes the time interval. By iteratively applying (6), we can estimate $\mathbf{q}_N$ at the last time step, which corresponds to the joint target $\mathbf{g}_q^*(\boldsymbol{\theta}) = \mathbf{q}_N$. Note that the forcing term $\mathbf{f}_t^p$ in (1) is a stochastic variable, and thus we use its expected values. With the new joint target, adapted joint trajectories can be generated using the probabilistic DMP described in (1), which will be exploited together with the adapted Cartesian trajectory to mimic human demonstrated trajectories in Cartesian and joint spaces as explained next.

B. Unifying Cartesian and Joint Constraints Using Null-space Exploration

Here we explain how to combine the adapted trajectories from Cartesian and joint spaces that were obtained as explained previously. On the basis of the new Cartesian target $\mathbf{g}_p^*$ and the retrieved distribution of the forcing term $\mathbf{f}_t^p \sim \mathcal{N}(\boldsymbol{\mu}_{f,t}^p, \boldsymbol{\Sigma}_{f,t}^p)$, we can derive the desired acceleration $\ddot{\bar{\boldsymbol{\xi}}}_t$ with mean $\mathbb{E}(\ddot{\bar{\boldsymbol{\xi}}}_t)$ and covariance $\mathbb{D}(\ddot{\bar{\boldsymbol{\xi}}}_t)$ by using (1) as

$$\begin{aligned}\mathbb{E}(\ddot{\bar{\boldsymbol{\xi}}}_t) &= \frac{1}{\tau^2}\left(\mathbf{K}^p(\mathbf{g}_p^* - \boldsymbol{\xi}_{t-1}) - \tau\mathbf{K}^v\dot{\boldsymbol{\xi}}_{t-1} + s\boldsymbol{\mu}_{f,t}^p\right)\\ \mathbb{D}(\ddot{\bar{\boldsymbol{\xi}}}_t) &= \frac{s^2}{\tau^4}\boldsymbol{\Sigma}_{f,t}^p\end{aligned}, \quad (7)$$

where $\boldsymbol{\xi}_{t-1}$ and $\dot{\boldsymbol{\xi}}_{t-1}$ respectively represent the current Cartesian position and velocity. From the acceleration $\ddot{\bar{\boldsymbol{\xi}}}_t$ computed in (7), the desired Cartesian velocity $\dot{\bar{\boldsymbol{\xi}}}_t$ and position $\bar{\boldsymbol{\xi}}_t$ can be computed by numerical integration. The transformed joint position $\bar{\mathbf{q}}_t$ estimated by using (6) has mean and covariance

$$\begin{aligned}\mathbb{E}(\bar{\mathbf{q}}_t^p|\boldsymbol{\theta}) &= \mathbf{q}_{t-1} + \mathbf{J}^\dagger(\dot{\boldsymbol{\xi}}_{t-1}\delta_t + \mathbb{E}(\ddot{\bar{\boldsymbol{\xi}}}_t)\delta_t^2) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\mathbf{N}(\boldsymbol{\theta})\delta_t\\ \mathbb{D}(\bar{\mathbf{q}}_t^p) &= \delta_t^4\mathbf{J}^\dagger\mathbb{D}(\ddot{\bar{\boldsymbol{\xi}}}_t)\mathbf{J}^{\dagger T}\end{aligned}. \quad (8)$$

The transformed joint trajectory $\bar{\mathbf{q}}_t$ actually corresponds to the adapted Cartesian trajectory $\bar{\boldsymbol{\xi}}_t$, which will be used in the hybrid learning process. Similarly, by using the new joint target $\mathbf{g}_q^*$ (estimated as described in Section III-A) and the retrieved distribution of the forcing term $\mathbf{f}_t^q \sim \mathcal{N}(\boldsymbol{\mu}_{f,t}^q, \boldsymbol{\Sigma}_{f,t}^q)$ at time t, we can estimate the adapted joint acceleration $\ddot{\bar{\mathbf{q}}}_t$ with mean and covariance

$$\begin{aligned}\mathbb{E}(\ddot{\bar{\mathbf{q}}}_t^q|\boldsymbol{\theta}) &= \frac{1}{\tau^2}\left(\mathbf{K}^p(\mathbf{g}_q^*(\boldsymbol{\theta}) - \mathbf{q}_{t-1}) - \tau\mathbf{K}^v\dot{\mathbf{q}}_{t-1} + s\boldsymbol{\mu}_{f,t}^q\right)\\ \mathbb{D}(\ddot{\bar{\mathbf{q}}}_t^q) &= \frac{s^2}{\tau^4}\boldsymbol{\Sigma}_{f,t}^q\end{aligned} \quad (9)$$

and its corresponding joint position $\mathbf{q}_t$ using numerical integration

$$\begin{aligned}\mathbb{E}(\bar{\mathbf{q}}_t^q|\boldsymbol{\theta}) &= \mathbf{q}_{t-1} + \dot{\mathbf{q}}_{t-1}\delta_t + \mathbb{E}(\ddot{\bar{\mathbf{q}}}_t^q|\boldsymbol{\theta})\delta_t^2\\ \mathbb{D}(\bar{\mathbf{q}}_t^q) &= \delta_t^4\mathbb{D}(\ddot{\bar{\mathbf{q}}}_t^q)\end{aligned}, \quad (10)$$

where $\mathbf{q}_{t-1}$ and $\dot{\mathbf{q}}_{t-1}$ represent the current joint position and velocity.

Since our goal is to mimic human demonstrations in both Cartesian and joint spaces, we need to unify the constraints extracted in both spaces (i.e., adapted trajectories). Therefore, it is desirable that the adapted joint state $\bar{\mathbf{q}}_t^q$ in (10) stays close to the transformed joint state $\bar{\mathbf{q}}_t^p$ in (8). In order to do so, we propose to minimize the objective function

$$J_e(\boldsymbol{\theta}) = \sum_{t=1}^{N} \mathbf{e}_t^\top\mathbb{D}(\bar{\mathbf{q}}_t^q)^{-1}\mathbf{e}_t, \quad (11)$$

with $\mathbf{e}_t = \mathbb{E}(\bar{\mathbf{q}}_t^q|\boldsymbol{\theta}) - \mathbb{E}(\bar{\mathbf{q}}_t^p|\boldsymbol{\theta})$. The covariance-weighted form of (11) ensures that $\mathbb{E}(\bar{\mathbf{q}}_t^q|\boldsymbol{\theta})$ precisely matches $\mathbb{E}(\bar{\mathbf{q}}_t^p|\boldsymbol{\theta})$

Algorithm 1 Hybrid trajectory optimization using null-space exploration

```
Learn $\mathcal{P}(s, \mathbf{f}^p)$ and $\mathcal{P}(s, \mathbf{f}^q)$ from demonstrations
Initialize $\mathbf{g}_p^*$, $\tau$, $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\Sigma}_\epsilon$, $c$, $\gamma_1$, $\gamma_2$ and $\gamma_3$
repeat
    for $h = 1$ to $H$ do
        Sample $\boldsymbol{\epsilon}_h \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon)$
        $\boldsymbol{\theta}_{n,h} \leftarrow \boldsymbol{\theta}^{(n)} + \boldsymbol{\epsilon}_h$
        for $t = 1$ to $N$ do
            Retrieve $\mathcal{N}(\boldsymbol{\mu}_{f,t}^p, \boldsymbol{\Sigma}_{f,t}^p)$ using GMR (2)
            Set the forcing term $\mathbf{f}_t^p = \boldsymbol{\mu}_{f,t}^p$
            Estimate $\ddot{\boldsymbol{\xi}}_t$ using (1) and $\{\boldsymbol{\xi}_t, \dot{\boldsymbol{\xi}}_t\}$
            Estimate $\mathbf{q}_t(\boldsymbol{\theta}_{n,h})$ using (6)
        end for
        $\mathbf{g}_q^* \leftarrow \mathbf{q}_N(\boldsymbol{\theta}_{n,h})$
        Set $J_e = J_l = J_s = 0$
        for $t = 1$ to $N$ do
            Retrieve $\mathcal{N}(\boldsymbol{\mu}_{f,t}^p, \boldsymbol{\Sigma}_{f,t}^p)$ and $\mathcal{N}(\boldsymbol{\mu}_{f,t}^q, \boldsymbol{\Sigma}_{f,t}^q)$
            Estimate $\mathbb{E}(\ddot{\boldsymbol{\xi}}_t)$ and $\mathbb{E}(\bar{\mathbf{q}}_t^p|\boldsymbol{\theta}_{n,h})$ with (7)-(8)
            Estimate $\mathbb{E}(\ddot{\bar{\mathbf{q}}}_t^q|\boldsymbol{\theta}_{n,h})$ and $\mathbb{D}(\ddot{\bar{\mathbf{q}}}_t^q)$ with (9)
            Estimate $\mathbb{E}(\bar{\mathbf{q}}_t^q|\boldsymbol{\theta}_{n,h})$ and $\mathbb{D}(\bar{\mathbf{q}}_t^q)$ with (10)
            $\mathbf{q}_t \leftarrow \bar{\mathbf{q}}_t^q$, $\dot{\mathbf{q}}_t \leftarrow \frac{1}{\delta_t}(\mathbf{q}_t - \mathbf{q}_{t-1})$
            $\mathbf{e}_t \leftarrow \mathbb{E}(\bar{\mathbf{q}}_t^q|\boldsymbol{\theta}_{n,h}) - \mathbb{E}(\bar{\mathbf{q}}_t^p|\boldsymbol{\theta}_{n,h})$
            Compute accumulated values $J_e$, $J_l$ and $J_s$
        end for
        Compute $J(\boldsymbol{\theta}_{n,h})$ using (14)
    end for
    Update $\boldsymbol{\theta}^{(n+1)}$ using (15)
until $\boldsymbol{\theta}$ converges
return $\{\mathbf{q}_t\}_{t=1}^N$
```

for the trajectory segment associated with small covariance $\mathbb{D}(\bar{\mathbf{q}}_t^q)$. Meanwhile, a low matching accuracy is allowed when the associated covariance $\mathbb{D}(\bar{\mathbf{q}}_t^q)$ is large. Note that this weighted scheme shares similarities with competitive imitation learning [5], minimum intervention control [9], prioritized control [6] and the trajectory similarity criterion [10].

Once the optimal $\boldsymbol{\theta}$ is computed, the joint state $\bar{\mathbf{q}}_t^q$ can be determined using (10), which is finally used for controlling the robot in joint space. Note that we can also consider the objective function as $\sum_{t=1}^N \mathbf{e}_t^\top \mathbb{D}(\bar{\mathbf{q}}_t^p)^{-1} \mathbf{e}_t$ and use $\bar{\mathbf{q}}_t^p$ as the joint command. However, due to possible inconsistencies between $\bar{\mathbf{q}}_t^p$ and $\bar{\mathbf{q}}_t^q$, we cannot strictly address both constraints simultaneously albeit that the covariance-weighted measure (11) is used to enforce both constraints to stay close. Therefore, we can select the one whose corresponding space plays a more significant role in the hybrid imitation learning, but this is achieved at the cost of loose imitation in the other space.

C. Additional Constraints and Reinforcement Learning of the Null-Space Parameter

Often, we might also need to consider additional constraints such as joint limits and motion smoothness, so that the robot can execute the desired joint trajectory safely [11]. In this context, let us first formulate the aforementioned constraints as two cost functions. The joint limit constraint is defined by[3]

$$J_l(\boldsymbol{\theta}) = \sum_{t=1}^{N} \sum_{k=1}^{\mathcal{D}} \left(\frac{q_{t,k} - m_k}{m_{u,k} - m_{l,k}} \right)^2, \tag{12}$$

where the subscript k represents the k-th joint, $m_{u,k}$ and $m_{l,k}$ respectively represent the upper and lower limits of the k-th joint q_k, and m_k denotes the middle value of the joint limits. In order to model the joint smoothness, we consider

$$J_s(\boldsymbol{\theta}) = \sum_{t=1}^{N} (\mathbf{q}_t - \mathbf{q}_{t-1})^\top (\mathbf{q}_t - \mathbf{q}_{t-1}). \tag{13}$$

Based on the above constraints, we can formulate a new objective (i.e., cost function) as follows

$$J(\boldsymbol{\theta}) = \gamma_1 J_e(\boldsymbol{\theta}) + \gamma_2 J_l(\boldsymbol{\theta}) + \gamma_3 J_s(\boldsymbol{\theta}) \tag{14}$$

with positive coefficients γ_1, γ_2 and γ_3.

Now, we optimize the null-space parameter $\boldsymbol{\theta}$ so as to minimize $J(\boldsymbol{\theta})$. Here, we apply a reward-weighted policy search method to address this optimization problem. The update rule of $\boldsymbol{\theta}$ is given by [12], [13]

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} + \frac{\sum_{h=1}^{H} \boldsymbol{\epsilon}_h e^{-cJ(\boldsymbol{\theta}^{(n)} + \boldsymbol{\epsilon}_h)}}{\sum_{h=1}^{H} e^{-cJ(\boldsymbol{\theta}^{(n)} + \boldsymbol{\epsilon}_h)}}, \tag{15}$$

where $c > 0$, $\boldsymbol{\theta}^{(n)}$ represents the null-space parameter at the n-th iteration, H denotes the number of roll-outs for each update and $\boldsymbol{\epsilon}_h \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon)$ represents an exploration noise. The entire proposed approach is summarized in *Algorithm* 1.

IV. Experiments

In this section, we evaluate the proposed method on the simulated and real COMAN humanoid robot [14] using a hand-shaking task which consists of a reaching movement and a waving motion (Fig. 3). This task requires a strong coordination between Cartesian and joint spaces in order to exhibit a natural hand-shaking movement. For the task we consider 10 degrees of freedom (DoF), 3 of which from the torso and 7 from the right arm.

We collected four robot hand-shaking trajectories in Cartesian and joint spaces simultaneously via kinesthetic teaching (as shown in the *first* row in Fig. 3). The demonstrated trajectories, as depicted in the *first* row in Fig. 4, are subsequently used for fitting the probabilistic DMPs. On the basis of the extracted phase variables and the corresponding forcing terms, we use a 4-states GMM to learn $\mathcal{P}(\mathbf{s}, \mathbf{f}^p)$ in Cartesian space and a 4-states GMM to learn $\mathcal{P}(\mathbf{s}, \mathbf{f}^q)$ in joint space. An illustration of the estimated joint probability distribution of $\{\mathbf{s}, f_1^p\}$ for the first Cartesian position component is shown in the *left* plot in Fig. 5, while the *right* plot corresponds to the seventh joint q_7, where the forcing terms with small covariances can be viewed as consistent dynamics when

[3] Note that the executed joint trajectory depends on $\mathbf{g}_q^*(\boldsymbol{\theta})$, thus this constraint is an implicit function of $\boldsymbol{\theta}$.

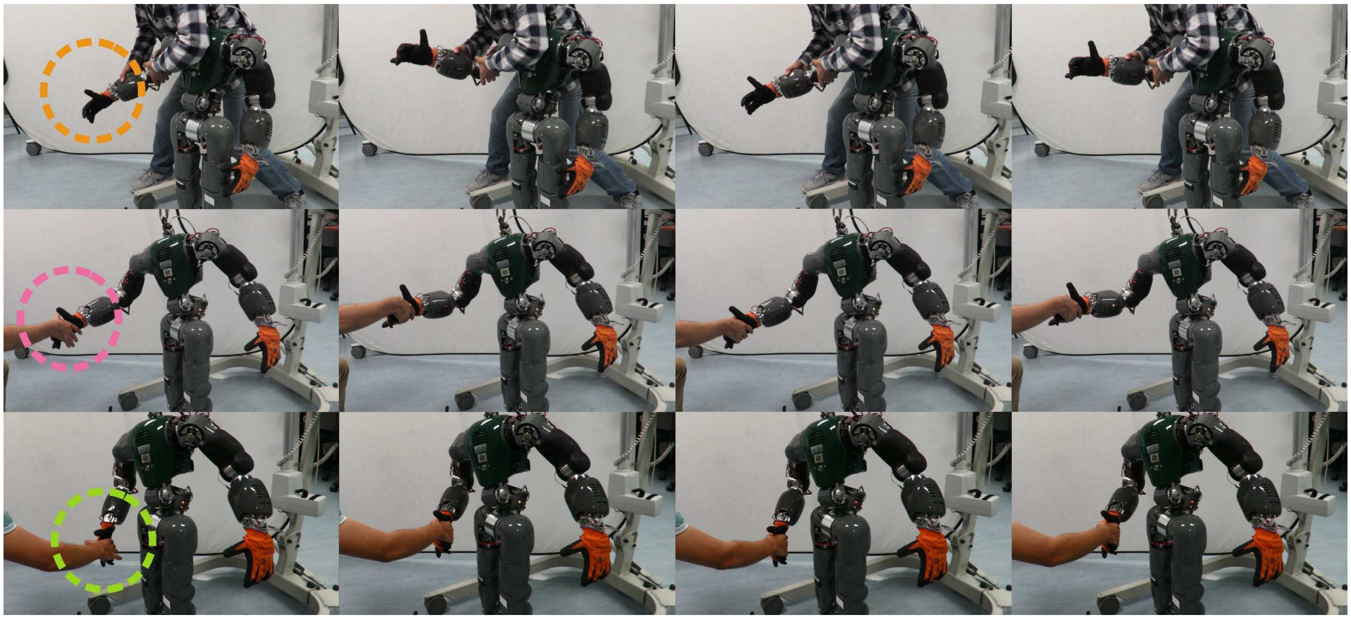
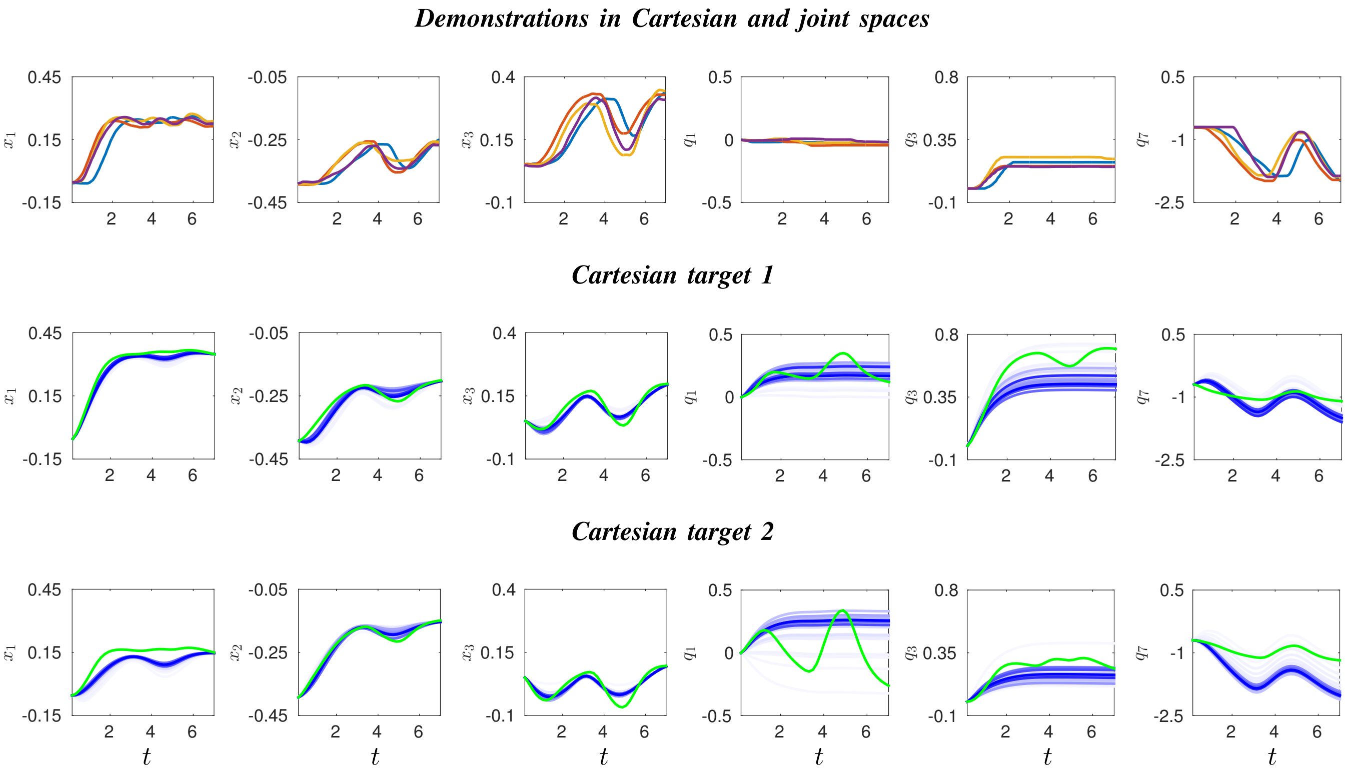


Fig. 3. Snapshots of human demonstrations (*first* row) in a hand-shaking task and adapted robot movement using the hybrid optimization approach (*second* and *third* rows). During the demonstration phase (*first* row), the human teacher moves the robot end-effector (right arm) to a specific Cartesian position and then performs a hand-shaking movement. During the generalization phase (*second* row and *third* row), given an unseen Cartesian target, the robot optimizes its trajectory to mimic demonstrated trajectories in both Cartesian and joint spaces. Dashed circles represent different areas for shaking hands.



Demonstrations in Cartesian and joint spaces

Cartesian target 1

Cartesian target 2

Fig. 4. Human demonstrations (*first* row) and adapted trajectories associated with new (unobserved) Cartesian targets (*second* and *third* rows), where the horizontal axis denotes time t. We only illustrate the trajectories of three Cartesian components (*left* three columns) and three robot joints (*right* three columns). The *second* and *third* rows respectively correspond to new Cartesian targets $[0.35\ \ -0.2\ 0.2]^\top$ and $[0.15\ \ -0.15\ 0.1]^\top$. The blue curves show the trajectory evolution based on the updated null-space parameter $\boldsymbol{\theta}$, where the color changing from light to dark denotes the learning evolution. The green curves correspond to the exclusive imitation learning in Cartesian space, where the Cartesian trajectory is adapted by using probabilistic DMP and the joint trajectories are determined by a typical Jacobian-based position controller without considering the null-space parameter.

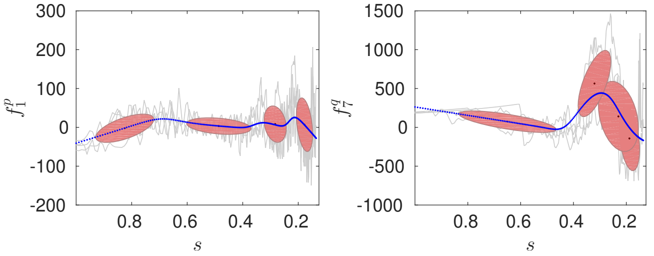


Fig. 5. Probability distributions of $\{\mathbf{s}, f_1^p\}$ (*left* graph) and $\{\mathbf{s}, f_7^q\}$ (*right* graph) encoded by GMM. Trajectories retrieved by GMR are depicted as blue curves. The gray trajectories represent the extracted data-pairs from demonstrations. The ellipses are Gaussian components in GMM.

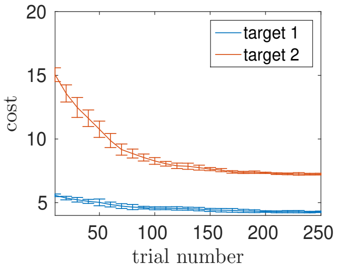


Fig. 6. Error-bar curves of cost values in the hand-shaking task. Solid curves represent mean values while vertical bars denote standard deviation.

modulating the spring-damper system in the probabilistic DMP. The relevant parameters in the *Algorithm* 1 are defined as $\tau = 7s$, $\boldsymbol{\theta}^{(0)} = \mathbf{0}$, $\boldsymbol{\Sigma}_\epsilon = 10^{-4}\mathbf{I}$, $c = 3$, $\gamma_1 = 0.01$, $\gamma_2 = 0.1$ and $\gamma_3 = 1$, where $\mathbf{I}$ is an identity matrix.

We separately test two Cartesian goals that are not observed in demonstrations $\mathbf{g}^*_{p_1} = [0.35 \ \ -0.2 \ \ 0.2]^\top$ and $\mathbf{g}^*_{p_2} = [0.15 \ \ -0.15 \ \ 0.1]^\top$ using the simulated COMAN robot. By employing our approach, the joint and Cartesian trajectories are adapted, as shown in the *second* and *third* rows in Fig. 4, where blue lines represent the evolution of Cartesian and joint space trajectories (from light to dark blue) as $\boldsymbol{\theta}$ is updated using (15). For comparison purposes, we also evaluate the imitation learning in the Cartesian space exclusively, where the Cartesian trajectory (green curves in Fig. 4) is generated by using the probabilistic DMP according to the Cartesian goals and meanwhile the Jacobian-based position controller (6) with $\boldsymbol{\theta} = \mathbf{0}$ is used to calculate the corresponding joint trajectories.

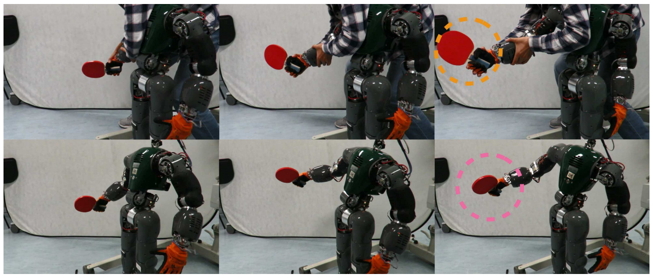


Fig. 7. Snapshots of human demonstration in a striking task (*top* row) and an adapted movement generated by the hybrid optimization method towards an unseen striking target. Dashed circles represent different striking targets.

From Fig. 4, we can observe that the proposed method is capable of generating trajectories (i.e., darkest blue curves in the *second* and *third* rows) that resemble human demonstrations (depicted in the *first* row) in both Cartesian and joint spaces. In contrast to our hybrid approach, the typical Jacobian-based position controller only mimic the demonstrated trajectories in the Cartesian space, while the shape of joint trajectories largely deviates from demonstrations, therefore failing to imitate the demonstrated posture patterns. Notice that, even though the trajectories resemble the demonstrations, occasionally minor deviations occur, which are due to the fact that trajectory constraints from Cartesian and joint spaces can sometimes conflict. Nevertheless, the reaching of the Cartesian goal is never compromised.

In order to statistically evaluate the performance of *Algorithm* 1, we run it 5 times for each Cartesian goal and for each run the null-space parameter is updated with 250 trials using (15). The statistical error-bar curves are illustrated in Fig. 6, showing that learning of the null-space parameter renders smaller cost values and thus the final joint trajectory is optimal in terms of task constraints in both Cartesian and joint spaces, motion smoothness and joint limits.

Finally, we test the proposed hybrid optimization on the real COMAN robot as well. Snapshots of the hand-shaking movement is provided in Fig. 3 (*second* and *third* rows). It is observed that the hand-shaking movement is mainly accomplished by the elbow joint (i.e. q_7) rather than other joints, which is closely similar to human demonstrated movements. Also, In order to show the applicability of the proposed method, we evaluate the proposed method in a different scenario (i.e., striking task), as shown in Fig. 7, proving the effectiveness of our approach again.

V. Related Work

The probabilistic DMP was applied in [15], where task parameters were incorporated so as to derive a task-parameterized DMP. In that work, the learning of trajectories was only carried out in either Cartesian or joint space, but without addressing the problem of learning a task with relevant Cartesian and joint constraints coming from the observed motion patterns.

Learning competing constraints in Cartesian and joint spaces was studied in [5], [6], where trajectories in both spaces were separately encoded using GMM, and subsequently the corresponding time-driven probabilistic reference trajectories were retrieved using GMR. In order to generalize the learned skill in Cartesian space, relative positions between the robot end-effector and the target were modeled instead of the absolute position values [5]. Later, a task-parameterized treatment of the trajectory adaptation in Cartesian space was proposed in [6], yielding reliable extrapolation capabilities in the Cartesian space. In order to combine trajectory constrains in Cartesian and joint spaces, both [5], [6] transformed the adapted Cartesian trajectory into joint space using the Jacobian-based position controller without exploiting the robot null-space.

Moreover, [5], [6] did not take into account the adaptation of joint trajectories. Namely, the distribution of demonstrated joint trajectories is directly applied to new situations without any modulation. Considering that a new Cartesian target might need joint trajectories that largely differ from the demonstrated ones in terms of joint movement ranges (e.g. q_1 in the *second* and *third* rows in Fig. 4 has significantly different values compared to the original ones in the *first* row), the distribution of demonstrated joint trajectories (particularly the mean values) becomes inappropriate, and hence the original joint constraints are undesirable.

In contrast to the exclusive adaptation in Cartesian space, we consider the trajectory adaptations in both the Cartesian and joint spaces simultaneously, allowing us to modulate constraints in both spaces for new situations, and thus generate trajectories that resemble human demonstrations. More specifically, unlike [5], [6], we exploit the null-space of the robot, rendering the exploitation of redundancy in inverse kinematics feasible. Also, note that [5], [6] calculate the final joint state as a Gaussian product of the transformed and demonstrated joint states, while we here exploit the covariance-weighted measure to make the transformed and adapted joint states stay close, providing a novel perspective to treat competing constraints in Cartesian and joint spaces in the context of imitation learning.

Another related work is proposed in [16], where both Cartesian and joint trajectories can be generated towards new targets by using a pure spring-damper system. However, this system can not encode human demonstrations and generalize the learned skills to new situations. Note that in [17] Cartesian trajectory is reproduced by using GMR while the joint trajectory is generated with the spring-damper model. Differing from [16], [17], we focus on the imitation learning (particularly from multiple demonstrations) and adaptations in both Cartesian and joint spaces, which allows the robot to learn various human skills straightforwardly.

VI. Conclusions and Future work

We introduced a hybrid trajectory optimization approach that can be employed to mimic human skills in both Cartesian and joint spaces simultaneously. Specifically, we proposed to exploit the robot null-space to explore solutions enforcing the robot to imitate human trajectories in both spaces. We demonstrated the effectiveness of the hybrid space optimization through hand-shaking and striking tasks.

In this paper, we considered the modeling of demonstrations without considering the external environment state. However, environment variables may be relevant in applications where the robot heavily interacts with its surroundings. For instance, in the robot table tennis setting, we need to determine which striking movement (e.g., forehand and backhand) is the most appropriate in order to return an incoming ball properly. Similar to the task-parameterized DMP [15] and the stylistic DMP with an additional movement descriptor [18], the extension of our work could incorporate the external state into the probabilistic DMP, so that the hybrid trajectory optimization is capable of choosing appropriate movement trajectories to learn and generalizing learned skills towards different stimuli more naturally.

In addition, the probabilistic DMP employs the same spring-damper formulation as DMP, which prevents its application to situations with velocity constraints. The recently developed non-parametric kernelized movement primitive (KMP) [19] allows for modulations of both position and velocity trajectories simultaneously. Thus, we plan to extend the hybrid optimization approach by exploiting KMP so as to handle various position and velocity requirements.

References

[1] B. D. Argall, S. Chernova and M. Veloso, “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469-483, 2009.

[2] Y. Huang, B. Schölkopf and J. Peters, “Learning optimal striking points for a ping-pong playing robot,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 4587-4592.

[3] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis and D. G. Caldwell, “Statistical dynamical systems for skills acquisition in humanoids,” in *Proc. IEEE International Conference on Humanoid Robots*, 2012, pp. 323-329.

[4] Y. Huang, D. Büchler, O. Koc, B. Schölkopf and J. Peters, “Jointly learning trajectory generation and hitting point prediction in robot table tennis,” in *Proc. IEEE International Conference on Humanoid Robots*, 2016, pp. 650-655.

[5] S. Calinon and A. Billard, “Statistical learning by imitation of competing constraints in joint space and task space,” *Advanced Robotics*, vol. 23, pp. 2059-2076, 2009.

[6] J. Silvério, S. Calinon, L. Rozo and D. G. Caldwell, “Learning competing constraints and task priorities from demonstrations of bimanual skills,” *arXiv*:1707.06791, 2017.

[7] T. Alizadeh, *Statistical learning of task modulated human movements through demonstration*, Italian Institute of Technology, PhD thesis, 2014.

[8] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor and S. Schaal, “Dynamical movement primitives: learning attractor models for motor behaviors,” *Neural Computation*, vol. 25, no. 2, pp. 328-373, 2013.

[9] S. Calinon, “A tutorial on task-parameterized movement learning and retrieval,” *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1-29, 2016.

[10] M. Muhlig, M. Gienger, S. Hellbach, J. J. Steil and C. Goerick, “Task-level imitation learning using variance-based movement optimization,” in *Proc. IEEE International Conference on Robotics and Automation*, 2009, pp. 1177-1184.

[11] M. Toussaint, M. Gienger, C. Goerick, “Optimization of sequential attractor-based movement for compact behaviour generation,” in *Proc. IEEE International Conference on Humanoid Robots*, 2007, pp. 122-129.

[12] F. Stulp and O. Sigaud, “Robot skill learning: from reinforcement learning to evolution strategies,” *Journal of Behavioral Robotics*, vol. 4, no. 1, pp. 49-61, 2013.

[13] E. Theodorou, J. Buchli and S. Schaal, “A generalized path integral control approach to reinforcement learning,” *Journal of Machine Learning Research*, vol. 11, pp. 3137-3181, 2010.

[14] N. G. Tsagarakis, S. Morfey and G. M. Cerda, “Compliant humanoid COMAN: optimal joint stiffness tuning for modal frequency control,” in *Proc. IEEE International Conference on Robotics and Automation*, 2013, pp. 673-678.

[15] A. Pervez and D. Lee, “Learning task-parameterized dynamic movement primitives using mixture of GMMs,“ *Intelligent Service Robotics* pp. 1-18, 2017.

[16] M. Hersch and A. G. Billard, “Reaching with multi-referential dynamical systems,” *Autonomous Robots*, vol. 25, pp. 71-83, 2008.

[17] M. Hersch, F. Guenter, S. Calinon and A. Billard, “Dynamical system modulation for robot learning via kinesthetic demonstrations,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463-1467, 2008.

[18] T. Matsubara, S. Hyon and J. Morimoto, “Learning stylistic dynamic movement primitives from multiple demonstrations,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1277-1283.

[19] Y. Huang, L. Rozo, J. Silvério and D. G. Caldwell, “Kernelized movement primitives,” *arXiv*:1708.08638, 2017.