2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Non-parametric Imitation Learning of Robot Motor Skills

Yanlong Huang[1], Leonel Rozo[2], João Silvério[1], and Darwin G. Caldwell[1]

Abstract— **Unstructured environments impose several challenges when robots are required to perform different tasks and adapt to unseen situations. In this context, a relevant problem arises: how can robots learn to perform various tasks and adapt to different conditions? A potential solution is to endow robots with learning capabilities. In this line, imitation learning emerges as an intuitive way to teach robots different motor skills. This learning approach typically mimics human demonstrations by extracting invariant motion patterns and subsequently applies these patterns to new situations. In this paper, we propose a novel kernel treatment of imitation learning, which endows the robot with imitative and adaptive capabilities. In particular, due to the kernel treatment, the proposed approach is capable of learning human skills associated with high-dimensional inputs. Furthermore, we study a new concept of correlation-adaptive imitation learning, which allows for the adaptation of correlations exhibited in high-dimensional demonstrated skills. Several toy examples and a collaborative task with a real robot are provided to verify the effectiveness of our approach.**

I. Introduction

From the perspective of trajectory generation, various robot skills can be accomplished by generating proper trajectories in either task or joint spaces. Trajectory generation for robots can be tackled from an imitation learning perspective [1], [2], [3], [4], [5], [6], where the robot learns the trajectory of interest from human demonstrations. Typically, the learned trajectories can be reproduced by the robot under conditions that are similar to those in which the demonstrations took place. However, the robot may also encounter unseen situations, such as obstacles and human intervention, which can be considered as new constraints of the task, requiring the robot to adapt its trajectory online to perform satisfactorily. Besides, unlike time-driven skills, many scenarios such as robot bi-manual operation and human-robot collaboration are often associated with high-dimensional inputs, which in turn increases the difficulties of robot skill learning and adaptation due to the complexity of input signals.

In order to make the imitation and adaptation of human skills feasible, many approaches have been developed, such as dynamic movement primitives (DMP) [2], Gaussian mixture model (GMM) [4], and probabilistic movement primitives (ProMP) [7]. Due to an explicit description of the trajectory dynamics, DMP introduces many open parameters in addition to basis functions and their weighting coefficients. Similarly, ProMP demands for a set of manually defined basis functions. Differing from DMP and ProMP, GMM has been employed to model the distribution of human demonstrations, which is exploited to retrieve desired trajectories through Gaussian mixture regression (GMR) [8]. Regarding robot adaptation, DMP is capable of generalizing trajectories towards new end-points while adaptation to varying via-points and velocity constraints are overlooked. ProMP formulates the modulation of trajectories as a Gaussian conditioning problem, and therefore provides an analytical solution to adapt trajectories towards new via-points or targets. The standard GMM/GMR framework does not offer adaptation features. A possible way to enhance the adaptation ability of DMP and GMM is using reinforcement learning (RL) [9], [10]. For instance, weighting exploration with returns [10] was employed to optimize the movement pattern of DMP. However, the model parameters of GMM commonly lie in a high-dimensional space (i.e., mixture coefficients, means and covariance matrices), and hence the re-optimization of GMM towards new requirements (e.g., via-points) is difficult. The same problem arises in DMP when the number of basis functions is large. In addition, the time-consuming learning process of RL might render the online adaptation impractical.

It is noteworthy that both DMP and ProMP are developed to learn time-driven skills, namely, demonstrated trajectories that depend on time. When we consider applications with high-dimensional inputs (e.g., human hand positions in human-robot collaboration), DMP and ProMP become less effective since a large number of basis functions are required to encapsulate the features of high-dimensional inputs, which is often referred to as the curse of dimensionality [11]. In this paper, we attempt to provide an alternative solution, which not only preserves the probabilistic properties exhibited in multiple demonstrations, but also deals with trajectory adaptations. More specifically, we aim to address the learning and adaption of demonstrated skills associated with high-dimensional inputs. Key features of our solution and the state-of-the-art methods are summarized in Table I.

Inspired by the kernel ridge regression (KRR) [12], [13] and its variant with a diagonal weighted scheme [14], [15], [16], we propose a novel *kernelized skill learning* approach that allows robots to learn probabilistic properties of multiple demonstrations (Section II-B), as well as modulate trajectories (Section II-C) when new task constraints arise on the fly (such as via-points or new target locations). The proposed solution is built on the well-established regression theory, rendering fewer open parameters and easy implementation. Furthermore, we study a new concept of *correlation-adaptive imitation learning* in Section II-D. With this new scheme, the coupling between high-dimensional motion variables can be

[1] Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genoa, Italy. yanlong.huang@iit.it; joao.silverio@iit.it;darwin.caldwell@iit.it

[2] Bosch Center for Artificial Intelligence, Renningen, Germany. leonel.rozo@de.bosch.com

This work was supported by the Italian Minister of Defense.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on May 19,2021 at 03:07:03 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON AMONG THE STATE-OF-THE-ART AND OUR APPROACH

	DMP	ProMP	GMM	Our Approach
Probabilistic	-	✓	✓	✓
Via–point	-	✓	-	✓
End–point	✓	✓	-	✓
High-dim Inputs	-	-	✓	✓

enforced or relaxed, allowing for a more flexible trajectory generation in terms of additional objective functions. We test the proposed kernelized approach in Section III, and discuss the related work in Section IV. Finally, we conclude our work in Section V.

II. NON-PARAMETRIC IMITATIVE SKILL LEARNING

In the context of imitation learning, an important observation is that the teacher often demonstrates skills differently even for the same task. Hence, the variability among demonstrations could be helpful as it encapsulates the important or consistent features of trajectories [17]. We first exploit the probabilistic properties from multiple human demonstrations (Section II-A), resulting in a trajectory distribution that we use to derive the non-parametric skill learning approach which we refer to as *kernelized movement primitives* (KMP) (Section II-B). Subsequently, on the basis of this approach, we study the trajectory adaptation (Section II-C) and the correlation-adaptive imitation learning (Section II-D).

A. Probabilistic Modeling of Demonstrated Skills

Assuming that we can access a set of demonstrated training data $\{\{\mathbf{s}_{n,h}, \boldsymbol{\xi}_{n,h}\}_{n=1}^{N}\}_{h=1}^{H}$, where $\mathbf{s}_{n,h} \in \mathbb{R}^{\mathcal{I}}$ is the input and $\boldsymbol{\xi}_{n,h} \in \mathbb{R}^{\mathcal{O}}$ denotes the output[1]. Here, the super-indexes $\mathcal{I}$, $\mathcal{O}$, H and N respectively represent the dimensionality of the input and output space, the number of demonstrations, and the trajectory length. In order to capture the probabilistic distribution of demonstrations, a number of algorithms can be employed, such as GMM [4], hidden Markov models [18] and kernel density estimation [19]. Let us take GMM as an example and employ it to encode the training data. More specifically, GMM is employed to estimate the joint probability distribution $\mathcal{P}(\mathbf{s}, \boldsymbol{\xi})$ from demonstrations, i.e., $\{\mathbf{s}, \boldsymbol{\xi}\} \sim \sum_l \pi_l \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$, where π_l, $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ respectively correspond to the prior probability, mean and covariance of the l-th Gaussian component. Furthermore, a *probabilistic reference trajectory* $\{\hat{\boldsymbol{\xi}}_n\}_{n=1}^{N}$ can be retrieved via GMR, where each point $\hat{\boldsymbol{\xi}}_n$ (associated with $\mathbf{s}_n$) is described by a conditional probability distribution $\hat{\boldsymbol{\xi}}_n|\mathbf{s}_n \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n)$ with mean $\hat{\boldsymbol{\mu}}_n$ and covariance $\hat{\boldsymbol{\Sigma}}_n$. This reference trajectory encapsulates the variability of demonstrations as well as the correlations among outputs.

For the sake of convenient description, we denote $\mathbf{D} = \{\mathbf{s}_n, \hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n\}_{n=1}^{N}$ as the *reference database*. In this paper, we exploit the probabilistic reference trajectory $\{\hat{\boldsymbol{\xi}}_n\}_{n=1}^{N}$ to derive KMP, whose formulation can be further used for modulating trajectories (Section II-C) and adapting correlations among high-dimensional motion variables (Section II-D).

B. Kernelized Skill Learning

Let us consider a *parametric trajectory*

$$\boldsymbol{\xi}(\mathbf{s}) = \boldsymbol{\Theta}(\mathbf{s})^{\top}\mathbf{w} \tag{1}$$

with the basis function matrix $\boldsymbol{\Theta}(\mathbf{s}) \in \mathbb{R}^{B\mathcal{O}\times\mathcal{O}}$ defined by

$$\boldsymbol{\Theta}(\mathbf{s}) = \begin{bmatrix} \boldsymbol{\varphi}(\mathbf{s}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}(\mathbf{s}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\varphi}(\mathbf{s}) \end{bmatrix}, \tag{2}$$

and the weight vector $\mathbf{w} \in \mathbb{R}^{B\mathcal{O}}$, where $\boldsymbol{\varphi}(\mathbf{s}) \in \mathbb{R}^{B}$ is a B-dimensional vector of basis functions. In order to incorporate the probabilistic reference trajectory (described in Section II-A) into this parametric representation, we propose to find a weight vector $\mathbf{w}$ such that $\{\boldsymbol{\Theta}(\mathbf{s}_n)^{\top}\mathbf{w}\}_{n=1}^{N}$ coincides with the reference database $\mathbf{D}$. This problem can be formulated as maximizing the posterior

$$J_p(\mathbf{w}) = \prod_{n=1}^{N} \mathcal{P}(\boldsymbol{\Theta}(\mathbf{s}_n)^{\top}\mathbf{w}|\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n). \tag{3}$$

Furthermore, by taking the logarithm of the posterior, this maximization (with respect to $\mathbf{w}$) is equivalent to minimizing a weighted sum of squared errors given by $\sum_{n=1}^{N}(\boldsymbol{\Theta}(\mathbf{s}_n)^{\top}\mathbf{w} - \hat{\boldsymbol{\mu}}_n)^{\top}\hat{\boldsymbol{\Sigma}}_n^{-1}(\boldsymbol{\Theta}(\mathbf{s}_n)^{\top}\mathbf{w} - \hat{\boldsymbol{\mu}}_n)$. In order to circumvent the over-fitting arising in this process, we introduce a penalty term $||\mathbf{w}||$. Thus, the resulting cost function to be minimized is

$$J(\mathbf{w}) = \sum_{n=1}^{N}(\boldsymbol{\Theta}(\mathbf{s}_n)^{\top}\mathbf{w} - \hat{\boldsymbol{\mu}}_n)^{\top}\hat{\boldsymbol{\Sigma}}_n^{-1}(\boldsymbol{\Theta}(\mathbf{s}_n)^{\top}\mathbf{w} - \hat{\boldsymbol{\mu}}_n) + \lambda\mathbf{w}^{\top}\mathbf{w}, \tag{4}$$

where $\lambda > 0$. This cost function shares the same formula with weighted least squares, except for the penalty term $\lambda\mathbf{w}^{\top}\mathbf{w}$. Also, it is similar to the common quadratic loss function minimized in KRR [12], [13], where $\hat{\boldsymbol{\Sigma}}_n^{-1} = \mathbf{I}_{\mathcal{O}}$ with $\mathbf{I}_{\mathcal{O}}$ representing the $\mathcal{O}$-dimensional identity matrix. However, we here exploit the variability of the demonstrations encapsulated in $\hat{\boldsymbol{\Sigma}}_n$ as an importance measure associated with each trajectory datapoint, which can be understood as relaxing or reinforcing the optimization for a particular datapoint. In other words, this variance-weighted cost function permits large deviations from the reference trajectory points with high variances, while demanding to be close when their associated variances are low. Note that similar variance-weighted strategies were studied in trajectory-GMM [4], linear quadratic regulators [20] and movement similarity criterion [21].

Through the dual transformation of KRR [12], [13], [16], the optimal solution $\widehat{\mathbf{w}}$ of (4) can be obtained, and subsequently for a new input $\mathbf{s}^{*}$ its corresponding output can be

[1] Note that the input $\mathbf{s}$ and output $\boldsymbol{\xi}$ can represent different types of variables. For instance, by considering $\mathbf{s}$ as the position of the robot and $\boldsymbol{\xi}$ as its velocity, the representation becomes an autonomous system formulation [4]. Alternatively, if $\mathbf{s}$ and $\boldsymbol{\xi}$ respectively represent time and position, the resulting encoding corresponds to a time-driven trajectory [7].

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on May 19,2021 at 03:07:03 UTC from IEEE Xplore. Restrictions apply.

written as

$$\boldsymbol{\xi}(\mathbf{s}^*) = \boldsymbol{\Theta}(\mathbf{s}^*)^\top \widehat{\mathbf{w}} = \boldsymbol{\Theta}(\mathbf{s}^*)^\top \boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \boldsymbol{\Sigma})^{-1} \boldsymbol{\mu}, \tag{5}$$

where

$$\begin{aligned} \boldsymbol{\Phi} &= [\boldsymbol{\Theta}(\mathbf{s}_1)\, \boldsymbol{\Theta}(\mathbf{s}_2) \cdots \boldsymbol{\Theta}(\mathbf{s}_N)], \\ \boldsymbol{\Sigma} &= blockdiag(\hat{\boldsymbol{\Sigma}}_1, \hat{\boldsymbol{\Sigma}}_2, \ldots, \hat{\boldsymbol{\Sigma}}_N), \\ \boldsymbol{\mu} &= [\hat{\boldsymbol{\mu}}_1^\top\, \hat{\boldsymbol{\mu}}_2^\top \cdots \hat{\boldsymbol{\mu}}_N^\top]^\top. \end{aligned} \tag{6}$$

Now, let us introduce the kernel function $k(\cdot,\cdot)$ and define $\boldsymbol{\varphi}(\mathbf{s}_i)^\top \boldsymbol{\varphi}(\mathbf{s}_j) = k(\mathbf{s}_i, \mathbf{s}_j)$. Then, we have

$$\mathbf{k}(\mathbf{s}_i, \mathbf{s}_j) = \boldsymbol{\Theta}(\mathbf{s}_i)^\top \boldsymbol{\Theta}(\mathbf{s}_j) = k(\mathbf{s}_i, \mathbf{s}_j) \mathbf{I}_{\mathcal{O}}. \tag{7}$$

Also, let us denote the matrix $\mathbf{K}$ with its block-component at i-th row and j-th column as $\mathbf{k}(\mathbf{s}_i, \mathbf{s}_j)$, then the prediction in (5) can be rewritten as

$$\boldsymbol{\xi}(\mathbf{s}^*) = \sum_i^N \mathbf{k}(\mathbf{s}^*, \mathbf{s}_i) \boldsymbol{\alpha}_i, \tag{8}$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^{\mathcal{O}}$ is the i-th component of the block matrix

$$\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^\top\, \boldsymbol{\alpha}_2^\top \cdots \boldsymbol{\alpha}_N^\top]^\top = (\mathbf{K} + \lambda \boldsymbol{\Sigma})^{-1} \boldsymbol{\mu}. \tag{9}$$

Note that the prediction described by (8) shares similarities with Gaussian process regression (GPR) [22], Heteroscedastic Gaussian processes (HGP) [14], [15] and cost regularized kernel regression (CrKR) [16]. If we replace $\hat{\boldsymbol{\Sigma}}_n$ in (4) by an identity matrix, the prediction (8) will become the estimated mean of GPR. Furthermore, if we use a diagonal weight matrix $\hat{\boldsymbol{\Sigma}}_n = c_n \mathbf{I}_{\mathcal{O}}$ instead, (8) is equivalent to the mean prediction of HGP and CrKR. However, all these approaches are derived in different contexts and thus have different application scenarios. Specifically, GPR, HGP and CrKR modeled target components separately without considering their correlations.

C. Trajectory Adaptation Using Kernelized Learning

While learning from demonstrations allows for accomplishing a specific task in the sense of reproduction, the adaptation ability is also pivotal, e.g., in dynamic and unstructured environments, where the robot needs to adapt its motions according to the external stimulus. To illustrate the importance of skill adaptation, let us consider a simple example: when an obstacle suddenly occupies an area that intersects the planned robot trajectory, the robot is therefore required to modulate its movement trajectory so as to avoid all the possible collisions. A similar modulation is necessary when the target location is varied over the course of the task.

We here tackle the problem of adapting trajectories to pass through new via-points/end-points by updating the reference database $\mathbf{D} = \{\mathbf{s}_n, \hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n\}_{n=1}^N$ used in (4) with the new desired points. More specifically, given M desired points $\{\bar{\mathbf{s}}_m, \bar{\boldsymbol{\xi}}_m\}_{m=1}^M$ defined by conditional probability distributions[2] $\bar{\boldsymbol{\xi}}_m | \bar{\mathbf{s}}_m \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_m, \bar{\boldsymbol{\Sigma}}_m)$, we can transform the imitation learning and trajectory adaptation into the problem of minimizing

$$J(\mathbf{w}) = \sum_{i=1}^{N+M} (\boldsymbol{\Theta}(\tilde{\mathbf{s}}_i)^\top \mathbf{w} - \tilde{\boldsymbol{\mu}}_i)^\top \tilde{\boldsymbol{\Sigma}}_i^{-1} (\boldsymbol{\Theta}(\tilde{\mathbf{s}}_i)^\top \mathbf{w} - \tilde{\boldsymbol{\mu}}_i) + \lambda \mathbf{w}^\top \mathbf{w}, \tag{10}$$

where $\{\tilde{\mathbf{s}}_i = \mathbf{s}_i, \tilde{\boldsymbol{\mu}}_i = \hat{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i = \hat{\boldsymbol{\Sigma}}_i\}, \forall i \in \{1, 2, \ldots, N\}$, and $\{\tilde{\mathbf{s}}_i = \bar{\mathbf{s}}_{i-N}, \tilde{\boldsymbol{\mu}}_i = \bar{\boldsymbol{\mu}}_{i-N}, \tilde{\boldsymbol{\Sigma}}_i = \bar{\boldsymbol{\Sigma}}_{i-N}\}, \forall i \in \{N+1, N+2, \ldots, N+M\}$. This new optimization problem essentially corresponds to the learning of an extended reference trajectory that consists of the original reference trajectory and various desired points. Similarly, it can be solved analytically, through (4)-(8). An algorithm of trajectory adaptation by using our approach is provided in Algorithm 1.

Algorithm 1 *Kernelized Trajectory Adaptation*

```
1: Initialization
     - Define the kernel $k(\cdot,\cdot)$ and set the factor $\lambda$.
     - Set desired points $\{\bar{\mathbf{s}}_m, \bar{\boldsymbol{\mu}}_m, \bar{\boldsymbol{\Sigma}}_m\}_{m=1}^M$.
2: Extract reference database (see Section II-A)
     - Collect demonstrations $\{\{\mathbf{s}_{n,h}, \boldsymbol{\xi}_{n,h}\}_{n=1}^N\}_{h=1}^H$.
     - Extract the reference database $\mathbf{D} = \{\mathbf{s}_n, \hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n\}_{n=1}^N$.
3: Incorporate desired points (see Section II-C)
     - Update $\mathbf{D}$ using desired points $\{\bar{\mathbf{s}}_m, \bar{\boldsymbol{\mu}}_m, \bar{\boldsymbol{\Sigma}}_m\}_{m=1}^M$.
4: Trajectory prediction (see Section II-B)
     - Input: query $\mathbf{s}^*$.
     - Output: $\boldsymbol{\xi}(\mathbf{s}^*) = \sum_i^N \mathbf{k}(\mathbf{s}^*, \mathbf{s}_i)\boldsymbol{\alpha}_i$
```

Note that conflicts might arise between the original reference database and the new desired points. Let us consider an extreme case: if $\bar{\mathbf{s}}_m = \mathbf{s}_n$ but $\bar{\boldsymbol{\mu}}_m$ is far away from $\hat{\boldsymbol{\mu}}_n$ while $\bar{\boldsymbol{\Sigma}}_m = \hat{\boldsymbol{\Sigma}}_n$, the resulting trajectory at the query point $\mathbf{s}_n$ would be a trade-off between $\bar{\boldsymbol{\mu}}_m$ and $\hat{\boldsymbol{\mu}}_n$, which hence fails to achieve our goal of trajectory adaptation. An alternative solution is to compare the similarities between the input variables $\{\bar{\mathbf{s}}_m\}_{m=1}^M$ with the inputs $\{\mathbf{s}_n\}_{n=1}^N$ of the reference database $\mathbf{D}$. For each $\bar{\mathbf{s}}_m$, if the distance between $\bar{\mathbf{s}}_m$ and its corresponding nearest input $\mathbf{s}_n$ is smaller than a predefined threshold, we replace $\{\mathbf{s}_n, \hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n\}$ in the reference database with $\{\bar{\mathbf{s}}_m, \bar{\boldsymbol{\mu}}_m, \bar{\boldsymbol{\Sigma}}_m\}$; otherwise, we insert $\{\bar{\mathbf{s}}_m, \bar{\boldsymbol{\mu}}_m, \bar{\boldsymbol{\Sigma}}_m\}$ into the reference database.

D. Correlation-adaptive Imitation Using Kernelized Learning

As done in (4), full covariance matrices $\{\hat{\boldsymbol{\Sigma}}_n\}_{n=1}^N$ (extracted from demonstrations) are employed in the optimization problem. These covariance matrices encode the correlation constraints between various motion variables. However, some inappropriate (e.g., too strong/weak) correlation constraints exhibited in the demonstrations might degrade the overall performance of the robot. Let us consider strong covariance matrices applied to a robot manipulation task with multiple joints: if a certain joint is suddenly damaged or perturbed, the movement of other joints will also be influenced heavily due to the strong constraints imposed by the covariance matrix. On the contrary, if the covariance matrices (i.e., correlations) are weak, the resulting trajectory might fail to resemble the demonstrated skills when external

[2] The distributions of desired points can be designed based on the new task requirements. For instance, for new via-points that the robot need to pass through precisely, small variances are assigned. On the contrary, via-points that allow for tracking errors, high variances can be defined.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on May 19,2021 at 03:07:03 UTC from IEEE Xplore. Restrictions apply.

perturbations are applied. Thus, a natural problem arises: can we relax or enforce the correlations among (the high-dimensional) motor variables so as to address unpredicted conditions or meet additional objectives?

Formally, we formulate the correlation-adaptive imitation learning as the problem of optimizing a diagonal matrix function $\mathbf{\Lambda}(\mathbf{s})$. To do so, we first reformulate our cost function (4) as follows

$$J(\mathbf{w};\mathbf{\Lambda}(\cdot))=\sum_{n=1}^{N}(\mathbf{\Theta}(\mathbf{s}_n)^{\top}\mathbf{w}-\hat{\boldsymbol{\mu}}_n)^{\top}(\mathbf{\Lambda}(\mathbf{s}_n)^{\top}\hat{\mathbf{\Sigma}}_n\mathbf{\Lambda}(\mathbf{s}_n))^{-1} (\mathbf{\Theta}(\mathbf{s}_n)^{\top}\mathbf{w}-\hat{\boldsymbol{\mu}}_n)+\lambda\mathbf{w}^{\top}\mathbf{w}, \qquad (11)$$

which now includes $\mathbf{\Lambda}(\cdot)$ as a term to adapt the correlation patterns of learned motor skills. For a given value of $\mathbf{\Lambda}(\cdot)$, this new objective function is first optimized as explained before (imitation/adaptation step), but now an additional objective $f(\{\boldsymbol{\xi}(\mathbf{s}_n^*)\}_{n=1}^{N})$ needs to be fulfilled[3], where $\{\mathbf{s}_n^*\}_{n=1}^{N}$ represents a sequence of new inquiries. We define a reward function $R(\mathbf{\Lambda}(\cdot))$ to measure how well $\mathbf{\Lambda}(\cdot)$ performs in terms of the objective function f. Thus, the correlation-adaptive imitation learning problem can be solved by finding $\mathbf{\Lambda}(\cdot)$ to maximize the reward $R(\mathbf{\Lambda}(\cdot))$, and accordingly many state-of-the-art RL algorithms can be used [23], [24]. Finally, the demonstrated covariance $\{\hat{\mathbf{\Sigma}}_n\}_{n=1}^{N}$ can be adjusted by $\{\mathbf{\Lambda}(\mathbf{s}_n)^{\top}\hat{\mathbf{\Sigma}}_n\mathbf{\Lambda}(\mathbf{s}_n)\}_{n=1}^{N}$, namely, the adaptation of correlations among the high-dimensional motor variables is accomplished. Therefore, the whole optimization process may be viewed as a first imitation/adaptation phase followed by a second model-free reinforcement learning phase.

III. Evaluations of the Approach

In this section, several examples of trajectory modulations and covariance adaptation are used to evaluate KMP. We first consider adapting trajectories to start-points/via-points/end-points (Section III-A). Then, we present an example (Section III-B) to illustrate the correlation-adaptive imitation learning. Finally, we carry out a collaborative hand task associated with a 6-D input on a real robot platform (Section III-C). More comprehensive evaluations of KMP can be found in our pre-print [17]. In the following evaluations, the Gaussian kernel $k(\mathbf{s}_i,\mathbf{s}_j)=exp(-\ell||\mathbf{s}_i-\mathbf{s}_j||^2)$ with $\ell>0$ is used.

A. Trajectory Modulation Examples

We study the trajectory adaptations on different 2-D hand-written letters. For each letter, five demonstrations comprising time t (i.e., input) and Cartesian position $[x(t)\,y(t)]^{\top}$ (i.e., output) are used to train a GMM (first column in Fig. 1), and subsequently a probabilistic reference trajectory is retrieved by GMR. This reference trajectory is used to initialize KMP. The relevant hyper-parameters are $\ell=8$ and $\lambda=0.1$. Figure 1 displays different trajectory modulation cases using our approach, showing that our approach successfully modifies the original reference trajectory to pass through various desired points (depicted by circles). For the sake of comparison, ProMP is also used as shown in Fig. 2. Note that even though ProMP performs similarly to our approach in the case of learning time-driven trajectories, its extension to learn trajectories with high-dimensional inputs becomes cumbersome due to the explicit definition of large number of basis functions. In contrast, our approach relies on the kernel instead of basis functions, allowing for learning demonstrations with high-dimensional input conveniently (see Section III-C).

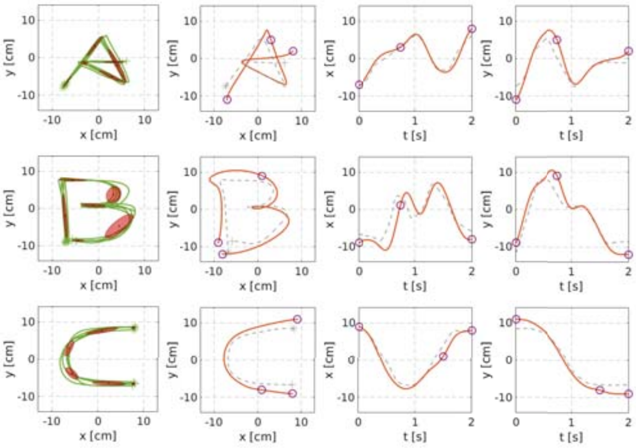


Fig. 1. Trajectory modulations using KMP on hand-written letters. *First column* shows the demonstrations of various hand-written letters and their corresponding GMM modeling, where the red solid ellipses represent Gaussian components. *Second–fourth columns* show the adapted trajectories (red solid lines) with different desired points (purple circles). The dashed gray curves depict the original reference trajectories, obtained as explained in Section II-A.

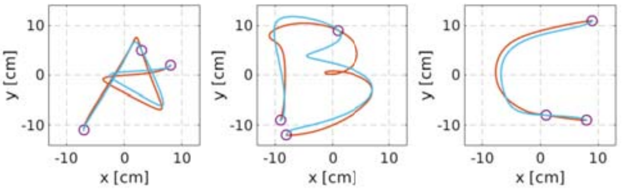


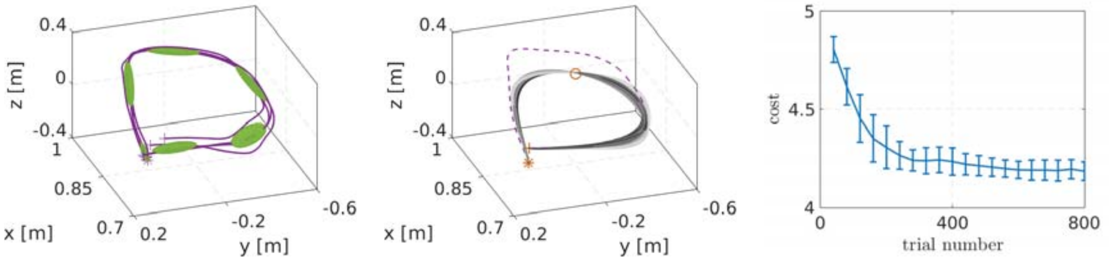
Fig. 2. Comparison of trajectory modulations by using KMP (red curves) and ProMP (blue curves), where circles denote desired points

B. Correlation-adaptive Imitation Learning Example

In this example, we consider the learning of 3-D trajectories ‘D’ using a simulated Barrett WAM robot, where four demonstrated trajectories are recorded, composed of time t (i.e., input) and Cartesian position $\boldsymbol{\xi}(t)=[x(t)\,y(t)\,z(t)]^{\top}$ (i.e., output), as shown in Fig. 3(*a*). Similarly to Section III-A, we use GMM to fit the demonstrations, and then extract the probabilistic reference trajectory so as to initialize KMP, where the hyperparameters are $\ell=0.1$ and $\lambda=2$. We define the reward as a function of a weighted joint smoothness. Specifically, we use Jacobian-based inverse kinematics to determine the corresponding joint trajectory, i.e., $\widehat{\mathbf{q}}_{t+1}=\mathbf{q}_t+\mathbf{J}(\mathbf{q}_t)^{\dagger}(\boldsymbol{\xi}_{t+1}-\boldsymbol{\xi}_t)$, where $\mathbf{J}^{\dagger}=\mathbf{J}^{\top}(\mathbf{J}\mathbf{J}^{\top})^{-1}$ with $\mathbf{J}$ being Jacobian matrix, $\mathbf{q}_t$ and $\widehat{\mathbf{q}}_{t+1}$ respectively denote the current joint position at time t and the desired joint position at time $t+1$. The reward is defined as $R=exp(-\gamma C)$, where $\gamma=5$

[3] This new objective function $f(\cdot)$ could, for instance, encompass movement smoothness and/or robot joint limits.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on May 19,2021 at 03:07:03 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 3. Evaluation of the correlation-adaptive imitation learning. (*a*) shows the fitting of demonstrated letters 'D' through GMM, where ellipsoids denote Gaussian components. (*b*) depicts the evolved process of adapted Cartesian trajectories (solid curves) with the color from light to dark showing the learning direction, where the circle denotes the desired via-point, '$*$' and '+' respectively represent the starting and ending points of trajectories. The dashed curve corresponds to the original reference trajectory. (*c*) shows the error-bar curve of the cost values over 10 runs.

and $C = \sum_{t=1}^{N-1} ||\mathbf{W}^{\frac{1}{2}}(\mathbf{q}_{t+1} - \mathbf{q}_t)||$ represents the cost of weighted joint smoothness with the weight matrix $\mathbf{W} > 0$.

Note that we also introduce a via-point constraint in this example. Namely, the resulting Cartesian trajectory should pass through a desired via-point and meanwhile its corresponding reward (defined in joint space) should be maximized. In order to find the optimal matrix function $\mathbf{\Lambda}(\cdot)$, we apply the policy improvement with path integrals algorithm with constant exploration [23], where we formulate $\mathbf{\Lambda}(t) = \boldsymbol{\phi}(t)^{\top}\boldsymbol{\theta}$ using a fixed Gaussian basis function matrix $\boldsymbol{\phi}(t)$ and the policy parameter $\boldsymbol{\theta}$ that needs to be learned. The evolved Cartesian trajectory is shown in Fig. 3(*b*) (solid curves), where the color from light to dark depicts the learning process. It can be seen that all the adapted Cartesian trajectories indeed pass through the desired via-point. The statistical analysis of cost values over 10 runs are plotted in Fig. 3(*c*), showing that the correlation adaptation indeed improves the weighted joint smoothness.

C. Collaborative Hand Task

Differing from learning various time-driven trajectories, we now consider a different task which requires a 6-D input, in particular a robot-assisted soldering scenario. As shown in Fig. 4, the task proceeds as follows: *(1)* the robot needs to hand over a circuit board to the user at the *handover location* $\mathbf{p}^h$ (Fig. 4*(b)*), where the user left-hand is used. *(2)* the user moves his left hand to place the circuit board at the *soldering location* $\mathbf{p}^s$ and simultaneously moves his right hand towards the soldering iron and then grasps it. Meanwhile, the robot is required to move towards the magnifying glass and grasp it at the *magnifying glass location* $\mathbf{p}^g$ (Fig. 4*(c)*). *(3)* the user moves his right hand to the soldering location so as to repair the circuit board. Meanwhile, the robot, holding the magnifying glass, moves towards the soldering place in order to allow the user to take a better look at the small components of the board (Fig. 4*(d)*).

Let us denote $\mathbf{p}^{\mathcal{H}_l}$, $\mathbf{p}^{\mathcal{H}_r}$ and $\mathbf{p}^{\mathcal{R}}$ as positions of the user left hand, right hand and robot end-effector (i.e., the "collaborative hand"), respectively. Since the robot is required to react properly according to the user hand positions, **we formulate the collaborative hand task as the prediction of the robot end-effector position according to the user hand positions (where time is not involved)**. In other words, in the prediction problem we consider $\mathbf{s} = \{\mathbf{p}^{\mathcal{H}_l}, \mathbf{p}^{\mathcal{H}_r}\}$ as the input (6-D) and $\boldsymbol{\xi}(\mathbf{s}) = \mathbf{p}^{\mathcal{R}}$ as the output (3-D). Following the procedure illustrated in Fig. 4, we collect five demonstrations comprising $\{\mathbf{p}^{\mathcal{H}_l}, \mathbf{p}^{\mathcal{H}_r}, \mathbf{p}^{\mathcal{R}}\}$ for training KMP, as shown in Fig. 5 (*a*). Note that the teacher only gets involved in the training phase. We fit the collected data using GMM, and subsequently extract a probabilistic reference trajectory using GMR, where the input for the probabilistic reference trajectory is sampled from the marginal probability distribution $\mathcal{P}(\mathbf{s})$, since in this scenario the exact input is unknown (unlike time t in previous experiments). The relevant hyperparameters are set to $\ell = 0.5$ and $\lambda = 2$.

Two evaluations are carried out to evaluate KMP in this scenario. First, we employ the learned reference database without adaptation so as to verify the reproduction ability of our approach. The user left- and right-hand trajectories as well as the real robot trajectory are plotted in Fig. 5 (*b*) (dotted curves), where the desired trajectory for robot end-effector is generated by our method. We can observe that the proposed method maintains the shape of the demonstrated trajectories for the robot while accomplishing the soldering task. Second, we evaluate the adaptation capability of KMP by varying the handover location $\mathbf{p}^h$, the magnifying glass location $\mathbf{p}^g$ as well as the soldering location $\mathbf{p}^s$. Note that these new locations are unseen in the demonstrations, thus we consider them as new via-point/end-point constraints. To take the handover as an example, we can define a via-point (associated with input) as $\{\bar{\mathbf{p}}_1^{\mathcal{H}_l}, \bar{\mathbf{p}}_1^{\mathcal{H}_r}, \bar{\mathbf{p}}_1^{\mathcal{R}}\}$, where $\bar{\mathbf{p}}_1^{\mathcal{H}_l} = \mathbf{p}^h$, $\bar{\mathbf{p}}_1^{\mathcal{H}_r} = \mathbf{p}_{ini}^{\mathcal{H}_r}$ and $\bar{\mathbf{p}}_1^{\mathcal{R}} = \mathbf{p}^h$, which implies that the robot should reach the new handover location $\mathbf{p}^h$ when the user left hand arrives at $\mathbf{p}^h$ and the user right hand stays at its initial position $\mathbf{p}_{ini}^{\mathcal{H}_r}$. Similarly, we can define additional via- and end-points to ensure that the robot grasps the magnifying glass at a new location $\mathbf{p}^g$ and assists the user at a new location $\mathbf{p}^s$. Thus, two via-points and one end-point are used to update the original reference database so as to address the three adaptation situations. Figure 5 (*b*) shows the adaptations of the robot trajectory (green solid curve) in accordance with the user hand trajectories (red and blue solid curves). It can be seen that the robot trajectory is indeed modulated towards the new handover, magnifying

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on May 19,2021 at 03:07:03 UTC from IEEE Xplore. Restrictions apply.

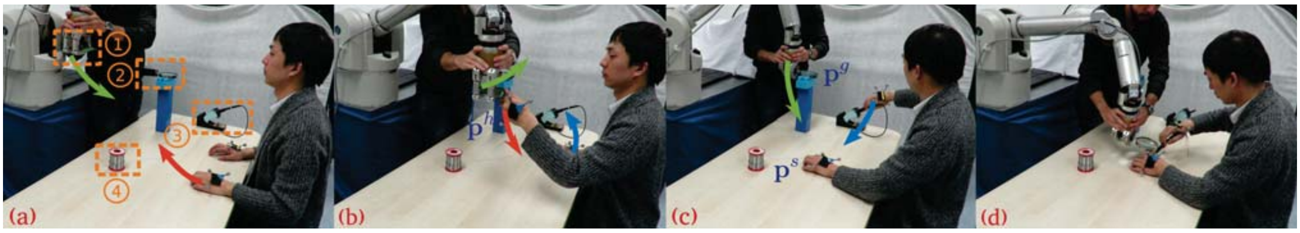
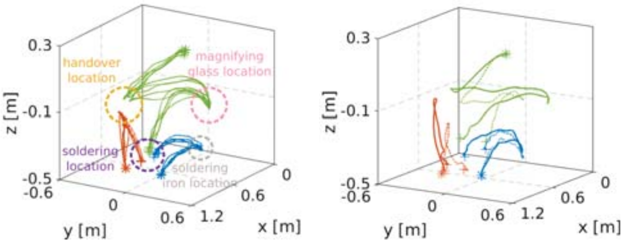


Fig. 4. The collaborative hand task in the soldering environment with the Barrett WAM robot. *(a)* shows the initial state of the user hands and the robot end-effector (i.e., the collaborative hand in this experiment). ①–④ separately correspond to the circuit board (held by the robot), magnifying glass, soldering iron and solder. *(b)* corresponds to the handover of the circuit board. *(c)* shows the robot grasping of the magnifying glass. *(d)* depicts the final scenario of the soldering task using both of the user hands and the robot end-effector. Red, blue and green arrows depict the movement directions of the user left hand, right hand and the robot end-effector, respectively.



(a) (b)

Fig. 5. Evaluations of the collaborative hand task. *(a)* shows demonstrations for the collaborative hand task, where the red and blue curves respectively correspond to the user left and right hands, while the green curves represent the demonstrated trajectories for the robot. The '$*$' and '+' mark the starting and ending points of various trajectories, respectively. *(b)* depicts the reproduction (dotted curves) and adaptation (solid curves) capabilities of our approach, where the user left-hand and right-hand trajectories (red and blue curves) are used to predict the robot end-effector trajectory (green curves).

glass and soldering locations, showing the capability of the proposed approach to adapt trajectories associated with high-dimensional inputs.

Note that the entire soldering task is accomplished without any trajectory segmentation for different subtasks, thus allowing for a straightforward learning of several sequential subtasks. Moreover, our approach makes the adaptation of learned skills associated with high-dimensional inputs feasible. Also, the fact that the learned KMP is driven by the user hand positions, allows for slower/faster hand movements since the prediction of KMP does not depend on time, hence alleviating the typical problems of time-alignment and phase-estimation in human-robot collaborations [25], [26], [27].

IV. Discussion

We here first compare KMP and ProMP [7]. For KMP, the joint distribution $\mathcal{P}(\mathbf{s},\boldsymbol{\xi})$ is first estimated from demonstrations, and subsequently a probabilistic reference trajectory $\{\hat{\boldsymbol{\xi}}_n\}_{n=1}^N$ with distribution $\hat{\boldsymbol{\xi}}_n|\mathbf{s}_n\sim\mathcal{N}(\hat{\boldsymbol{\mu}}_n,\hat{\boldsymbol{\Sigma}}_n)$ is retrieved. The imitation learning is formulated as an optimization problem (as described in (4)) where an optimal $\mathbf{w}$ is derived, which maximizes the posterior $\prod_{n=1}^N\mathcal{P}(\boldsymbol{\Theta}(\mathbf{s}_n)^\top\mathbf{w}|\hat{\boldsymbol{\mu}}_n,\hat{\boldsymbol{\Sigma}}_n)$. In contrast, ProMP estimates the distribution over weights $\mathcal{P}(\mathbf{w})$, i.e., $\mathbf{w}\sim\mathcal{N}(\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)$, which maximizes the likelihood $\prod_{h=1}^H\prod_{n=1}^N\mathcal{P}(\boldsymbol{\xi}_{n,h}|\boldsymbol{\Theta}(\mathbf{s}_{n,h})^\top\boldsymbol{\mu}_w,\boldsymbol{\Theta}(\mathbf{s}_{n,h})^\top\boldsymbol{\Sigma}_w\boldsymbol{\Theta}(\mathbf{s}_{n,h}))$. Then, the optimal distribution of $\mathbf{w}$ described by its mean $\boldsymbol{\mu}_w$ and variance $\boldsymbol{\Sigma}_w$ is found. To solve this maximization problem for ProMP, for each demonstration regularized least-squares can be used to estimate its movement pattern vector $\mathbf{w}$ [28], where basis functions are used to fit these demonstrations. Subsequently, with movement patterns from all demonstrations, the distribution $\mathcal{P}(\mathbf{w})$ is estimated. A direct problem in ProMP comes up with fixed basis functions, which suffers from the curse of dimensionality. In contrast, our approach is combined with a kernel function, alleviating the need for basis functions. The other problem in ProMP is the estimation of $\mathcal{P}(\mathbf{w})$. If the dimension of $\mathbf{w}$ (i.e., $B\mathcal{O}$) is too large compared to the number of demonstrations H, the estimated covariance $\boldsymbol{\Sigma}_w$ may be singular. In contrast, our approach needs a probabilistic reference trajectory, which is derived from the probability distribution of $\{\mathbf{s},\boldsymbol{\xi}\}$ that has a lower dimension (i.e., $\mathcal{I}+\mathcal{O}$).

Other related works were reported in [29], [30], where imitation learning and motion planning were formulated into a single framework and the entire sequence of trajectory points was determined from an optimization perspective. However, this kind of optimization may become inefficient when the trajectory length is quite large. Moreover, as pointed out in [31], [32], an additional interpolation is often required since the query points might be different from the predefined ones. Specifically, in comparison with our work, both [29] and [30] focus on learning time-driven trajectories without addressing the problem of learning trajectories associated with high-dimensional inputs.

V. Conclusions

We have shown a novel non-parametric imitation learning approach and its applications in trajectory modulations. Also, we studied a new concept of correlation-adaptive imitation learning, allowing for the adaptation of correlations among demonstrated high-dimensional motion variables, given additional performance requirements. Since our approach employs the kernel treatment instead of the explicit basis functions, it enables the convenient extension to the learning of complex and high-dimensional trajectories. In the future, we plan to apply our approach to the concurrent imitation learning in Cartesian space and joint space [33], [34].

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on May 19,2021 at 03:07:03 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *Proc. International Conference on Machine Learning*, 1997, pp. 12-20.

[2] 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.

[3] 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.

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

[5] J. Silvério, L. Rozo, S. Calinon and D. G. Caldwell, "Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 464-470.

[6] Y. Huang, J. Silvério, L. Rozo, and D. G. Caldwell, "Generalized task-parameterized skill learning," in *Proc. IEEE International Conference on Robotics and Automation*, 2018, pp. 5667-5674.

[7] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Proc. Advances in Neural Information Processing Systems*, 2013, pp. 2616-2624.

[8] D. A. Cohn, Z. Ghahramani, M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 129-145, 1996.

[9] J. Buchli, F. Stulp, E. Theodorou and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 820-833, 2011.

[10] J. Kober and J. Peters, "Policy search for motor primitives in robotics," in *Proc. Advances in Neural Information Processing Systems*, 2009, pp. 849-856.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, Chapter 3, Springer, 2006.

[12] C. Saunders, A. Gammerman and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proc. International Conference on Machine Learning*, 1998, pp. 515-521.

[13] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Chapter 14.4.3, pp. 492-493, The MIT Press, 2012.

[14] P. W. Goldberg, C. K. Williams and C. M. Bishop, "Regression with input-dependent noise: a Gaussian process treatment," in *Proc. Advances in Neural Information Processing Systems*, 1998, pp. 493-499.

[15] K. Kersting, C. Plagemann, P. Pfaff and W. Burgard, "Most likely heteroscedastic Gaussian process regression," in *Proc. International Conference on Machine learning*, 2007, pp. 393-400.

[16] J. Kober, E. Öztop and J. Peters, "Reinforcement learning to adjust robot movements to new situations," in *Proc. International Joint Conference on Artificial Intelligence*, 2011, pp. 2650-2655.

[17] Y. Huang, L. Rozo, J. Silvério and D. G. Caldwell. "Kernelized movement primitives," *arXiv*:1708.08638v1,v2, 2017.

[18] D. Kuli and N. Yoshihiko, "Incremental learning of human behaviors using hierarchical hidden markov mdels," in *Proc. IEEE International Conference on Intelligent Robots and Systems*, 2010, pp. 4649-4655.

[19] M. P. Holmes, A. G. Gray and C. L. Isbell, "Fast nonparametric conditional density estimation," in *Proc. Uncertainty in Artificial Intelligence*, 2007, pp. 175-182.

[20] J. R. Medina, D. Lee and S. Hirche, "Risk-sensitive optimal feedback control for haptic assistance," in *Proc. IEEE International Conference on Robotics and Automation*, 2012, pp. 1025-1031.

[21] 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.

[22] C. E. Rasmussen and C. K. William, *Gaussian Processes for Machine Learning*, Chapter 2, Cambridge: MIT press, 2006.

[23] 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.

[24] G. Lever and R. Stafford, "Modelling policies in MDPs in reproducing kernel hilbert space," *Artificial Intelligence and Statistics*, pp. 590-598, 2015.

[25] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *Proc. IEEE International Conference on Robotics and Automation*, 2014, pp. 2831-2837.

[26] G. Maeda, M. Ewerton, G. Neumann, R. Lioutikov and J. Peters, "Phase estimation for fast action recognition and trajectory generation in humanrobot collaboration," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1579-1594, 2017

[27] Y. Cui, J. Poon, J. V. Miroz, K. Yamazaki, K. Sugimoto and T. Matsubara, "Environment-adaptive interaction primitives through visual context for humanrobot motor skill learning," *Autonomous Robots*, pp. 1-16, 2018.

[28] D. Koert, G. Maeda, R. Lioutikov, G. Neumann and J. Peters, "Demonstration based trajectory optimization for generalizable robot motions," in *Proc. IEEE International Conference on Humanoid Robots*, 2016, pp. 515-522.

[29] T. Osa, A. M. Esfahani, R. Stolkin, R. Lioutikov, J. Peters J and G. Neumann, "Guiding trajectory optimization by demonstrated distributions," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp.819-826, 2017.

[30] M. A. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova and B. Boots, "Towards robust skill generalization: unifying learning from demonstration and motion planning," in *Proc. 1st Conference on Robot Learning*, 2017.

[31] M. Mukadam, X. Yan and B. Boots, "Gaussian process motion planning," in *Proc. IEEE International Conference on Robotics and Automation*, 2016, pp. 9-15.

[32] J. Dong, M. Mukadam, F. Dellaert and B. Boots, "Motion planning as probabilistic inference using Gaussian processes and factor graphs," in *Proc. Robotics: Science and Systems*, 2016.

[33] Y. Huang, J. Silvério, L. Rozo, and D. G. Caldwell, "Hybrid probabilistic trajectory optimization using null-space exploration," in *Proc. IEEE International Conference on Robotics and Automation*, 2018, pp. 7226-7232.

[34] J. Silvério, Y. Huang, L. Rozo, S. Calinon, and D. G. Caldwell, "Probabilistic learning of torque controllers from kinematic and force constraints," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 6552-6559.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on May 19,2021 at 03:07:03 UTC from IEEE Xplore. Restrictions apply.