Uncertainty-Aware Imitation Learning using Kernelized Movement Primitives

Preprint · March 2019

5 authors, including:

João Silvério
Idiap Research Institute
23 PUBLICATIONS   **206** CITATIONS

SEE PROFILE

Yanlong Huang
University of Leeds
25 PUBLICATIONS   **193** CITATIONS

SEE PROFILE

Fares J. Abu-Dakka
Aalto University
32 PUBLICATIONS   **176** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project PAPREC: Automatic grasping of piled parts from unstructured container View project

Project PHOLUS View project

Uncertainty-Aware Imitation Learning using Kernelized Movement Primitives

João Silvério[1,2], Yanlong Huang[2], Fares J. Abu-Dakka[2,3], Leonel Rozo[4] and Darwin G. Caldwell[2]

Abstract—During the past few years, probabilistic approaches to imitation learning have earned a relevant place in the robotics literature. One of their most prominent features is that, in addition to extracting a mean trajectory from task demonstrations, they provide a variance estimation. The intuitive meaning of this variance, however, changes across different techniques, indicating either *variability* or *uncertainty*. In this paper we leverage kernelized movement primitives (KMP) to provide a new perspective on imitation learning by predicting variability, correlations and uncertainty using a single model. This rich set of information is used in combination with the *fusion of optimal controllers* to learn robot actions from data, with two main advantages: i) robots become safe when uncertain about their actions and ii) they are able to leverage partial demonstrations, given as elementary sub-tasks, to optimally perform a higher level, more complex task. We showcase our approach in a painting task, where a human user and a KUKA robot collaborate to paint a wooden board. The task is divided into two sub-tasks and we show that the robot becomes compliant (hence safe) outside the training regions and executes the two sub-tasks with optimal gains otherwise.
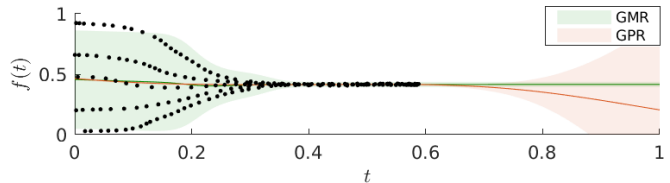
Fig. 1: Gaussian mixture regression (GMR) and Gaussian process regression (GPR) provide complementary notions of variance (represented as green and red shaded areas) as variability and absence of training datapoints (depicted as black dots). With a unified technique, robots can learn controllers that are modulated by both types of information.

I. Introduction

Probabilistic approaches to imitation learning [1] have witnessed a rise in popularity during the past few years. They are often seen as complementing deterministic techniques, such as dynamic movement primitives [2], with more complete descriptions of demonstration data, in particular in the form of covariance matrices that encode both the variability and correlations in the data. Widely used approaches at this level include Gaussian mixture models (GMM), popularized by the works of Calinon (e.g. [3]) and more recently, probabilistic movement primitives [4] and kernelized movement primitives (KMP) [5].

In recent work [6], [7], we discussed a fundamental difference between the type of variance encapsulated by the predictions of classical probabilistic techniques, particularly Gaussian mixture regression (GMR) and Gaussian process regression (GPR) [8]. We showed that the variance predicted by these two techniques has distinct, complementary interpretations. In particular that GMR predictions measure the *variability* in the training data, while those of GPR quantify the degree of *uncertainty*, increasing as one queries a model farther away from the region where it was trained. These properties are illustrated in Fig. 1. This finding led us to inquire: is there a probabilistic technique that can simultaneously predict *both variability and uncertainty*? Are these two notions compatible and unifiable into a single imitation learning framework where they both provide clear advantages from a learning perspective? In this paper we try to answer these questions.

The two types of variance have been individually leveraged by different lines of work. For instance, variability and data correlations (encapsulated in full covariance matrices) have been used to modulate control gains in several works [3], [9], [10], [11]. Uncertainty, in the sense of absence of data/information, is also a concept with tradition in robotics. Problems in robot localization [12], control [13] and, more recently, Bayesian optimization [14], leverage uncertainty information to direct the robot to optimal performance. In [7] we took advantage of uncertainty to regulate robot stiffness, in order to make it compliant (and safer) when uncertain about its actions. However, to the best of our knowledge, variability and uncertainty have never been exploited simultaneously in imitation learning.

In this paper we introduce an approach that predicts *variability*, *correlations* and *uncertainty* from KMP and uses this information to design optimal controllers from demonstrations. These drive the robot with high precision when the variability in the data is low (while respecting the observed correlations across degrees of freedom) and render the robot compliant (and safer to interact with) when the uncertainty is high. The uncertainty is further leveraged by the robot to know when different controllers, responsible for the execution of separate, elementary sub-tasks, should be activated. In particular we:

[1]Idiap Research Institute, CH-1920 Martigny, Switzerland (e-mail: joao.silverio@idiap.ch).

[2]Department of Advanced Robotics, Istituto Italiano di Tecnologia, 16163 Genova, Italy (e-mail: name.surname@iit.it)

[3]Intelligent Robotics Group, EEA, Aalto University, FI-00076 Aalto, Finland (e-mail: fares.abu-dakka@aalto.fi).

[4]Bosch Center for Artificial Intelligence, 71272 Renningen, Germany (e-mail:leonel.rozo@de.bosch.com).

1) demonstrate that KMP predicts full covariance matrices and uncertainty (Sections III and IV-A)
2) exploit a linear quadratic regulator (LQR) formulation that yields control gains which are a function of both covariance *and* uncertainty (Section IV-B)
3) dovetail 1), 2) with the concept of *fusion of controllers* [6] which allows for demonstrating one complex task as separate sub-tasks, whose activation depends on individual uncertainty levels (Section IV-C)

Experimentally, we expand on a previously published robot-assisted painting scenario and validate the approach using a KUKA LWR where different types of controllers are used for individual sub-tasks (Section V). We provide a discussion of the approach and the obtained results in Section VI and concluding remarks and possible extensions in Section VII.

II. RELATED WORK

Most probabilistic regression techniques provide variance predictions in some form. GMR, relying on a previously trained GMM, computes full covariance matrices encoding the correlation between output variables. However, it does not measure uncertainty, defaulting to the covariance of the closest Gaussian component when a query point is far from the model. GPR, despite estimating uncertainty, assumes constant noise therefore not taking the variability of the outputs into account. Heteroscedastic Gaussian Processes (HGP) [15], [16] introduce an input-dependent noise model into the regression problem. Nonetheless, tasks with multiple outputs require the training of separate HGP models, thus output correlations are not straightforward to learn in the standard formulation. In addition, the noise is treated as a latent function, hence each HGP depends upon the definition of two Gaussian processes (GP) per output, scaling poorly with the number of outputs. In [17], Choi *et al.* propose to use mixture density networks (MDN) in an imitation learning context to predict both variability and uncertainty. The main drawback of the approach, similarly to HGP, is that outputs are assumed to be uncorrelated. Moreover, in [17] only the uncertainty is used in the proposed imitation learning framework, without considering variability. As opposed to the aforementioned works, we here show that KMP predicts both full covariance matrices and a diagonal uncertainty matrix, parameterized by its hyperparameters, allowing the access to all the desired information. Table I details the differences between variance predictions of different algorithms, highlighting that KMP estimates all desired features in our approach.

In terms of estimating optimal controllers from demonstrations, previous works have either exploited full covariance matrices encoding variability and correlations [3], [9], [10] or diagonal uncertainty matrices [7]. While the former are aimed at control efficiency, by having the robot apply higher control efforts where required (depending on variability), the latter target safety, with the robot becoming more compliant when uncertain about its actions. The LQR we propose in Section IV-B is identical to the one in [3], [7], [11]. However, by benefiting from the KMP predictions, it unifies the best

| | Types of prediction | | |
	Variability	Uncertainty	Correlations
GMM/GMR [3]	✓	–	✓
GPR [8]	–	✓	–
HGP [15], [16]	✓	✓	–
MDN [17]	✓	✓	–
Our approach	✓	✓	✓

TABLE I: (Co)variance predictions of different techniques.

of the two approaches. Umlauft *et al.* [18] propose a related formulation where, using Wishart processes, they build full covariance matrices with uncertainty. However their solution requires a very high number of parameters, whose estimation relies heavily on optimization, and their control gains are set heuristically.

Finally, inspired by [19], in [6] we proposed a *fusion of controllers* to allow robots to smoothly switch between sub-tasks based on the uncertainty of each sub-task's controller. Here we go one step further and consider *optimal* controllers learned from demonstrations into the fusion, instead of manually defining the control gains. In previous work [11], we have studied the fusion of optimal controllers. However, in that case we focused on time-driven trajectories whereas here we consider multi-dimensional inputs and uncertainty.

The approach described in the next sections therefore aims at a seamless unification of concepts exploited in previous work, taking imitation learning one step ahead into the learning of optimal controllers for potentially complex tasks.

III. KERNELIZED MOVEMENT PRIMITIVES

We consider datasets comprised of H demonstrations with length T, $\{\{\boldsymbol{\xi}_{\mathcal{I}}^{t,h}, \boldsymbol{\xi}_{\mathcal{O}}^{t,h}\}_{t=1}^{T}\}_{h=1}^{H}$ where $\boldsymbol{\xi}_{\mathcal{I}} \in \mathbb{R}^{D_{\mathcal{I}}}$ and $\boldsymbol{\xi}_{\mathcal{O}} \in \mathbb{R}^{D_{\mathcal{O}}}$ denote inputs and outputs ($\mathcal{I}, \mathcal{O}$ are initials for 'input' and 'output'), respectively, and $D_{\mathcal{I}}, D_{\mathcal{O}}$ are their dimensions. $\boldsymbol{\xi}_{\mathcal{I}}$ can represent any variable of interest to drive the movement synthesis (e.g., time, object/human poses) and $\boldsymbol{\xi}_{\mathcal{O}}$ encodes the desired state of the robot (e.g., an end-effector position, a joint space configuration). KMP assumes access to an N-dimensional probabilistic trajectory distribution $\{\boldsymbol{\xi}_{\mathcal{I}}^{n}, \hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n\}_{n=1}^{N}$ mapping a sequence of inputs to their corresponding means and covariances, which encompass the important features in the demonstration data. This probabilistic reference trajectory can be obtained in various ways, for example by computing means and covariances empirically at different points in a dataset or by using unsupervised clustering techniques. Here we follow the latter direction, in particular by using a GMM to cluster the data and GMR to obtain the trajectory distribution that initializes KMP (done once after data collection).

By concatenating the trajectory distribution into $\boldsymbol{\mu} = [\hat{\boldsymbol{\mu}}_1^\top \ \dots \ \hat{\boldsymbol{\mu}}_N^\top]^\top$ and $\boldsymbol{\Sigma} = \text{blockdiag}(\hat{\boldsymbol{\Sigma}}_1, \dots, \hat{\boldsymbol{\Sigma}}_N)$, KMP predicts a new Gaussian distribution at new test points $\boldsymbol{\xi}_{\mathcal{I}}^*$ according to [5]

$$\boldsymbol{\mu}_{\mathcal{O}}^* = \boldsymbol{k}^* (\boldsymbol{K} + \lambda_1 \boldsymbol{\Sigma})^{-1} \boldsymbol{\mu}, \tag{1}$$

$$\boldsymbol{\Sigma}_{\mathcal{O}}^* = \frac{N}{\lambda_2} \left(\boldsymbol{k}^{**} - \boldsymbol{k}^* (\boldsymbol{K} + \lambda_2 \boldsymbol{\Sigma})^{-1} \boldsymbol{k}^{*\top} \right). \tag{2}$$

where

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^1, \boldsymbol{\xi}_{\mathcal{I}}^1) & \cdots & \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^1, \boldsymbol{\xi}_{\mathcal{I}}^N) \\ \vdots & \ddots & \vdots \\ \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^N, \boldsymbol{\xi}_{\mathcal{I}}^1) & \cdots & \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^N, \boldsymbol{\xi}_{\mathcal{I}}^N) \end{bmatrix} \quad (3)$$

is a matrix evaluating a chosen kernel function $k(.,.)$ at the training inputs, $\boldsymbol{k}^* = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^*, \boldsymbol{\xi}_{\mathcal{I}}^1) \dots \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^*, \boldsymbol{\xi}_{\mathcal{I}}^N) \end{bmatrix}$ and $\boldsymbol{k}^{**} = \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^*, \boldsymbol{\xi}_{\mathcal{I}}^*)$. Moreover, $\boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^i, \boldsymbol{\xi}_{\mathcal{I}}^j) = k(\boldsymbol{\xi}_{\mathcal{I}}^i, \boldsymbol{\xi}_{\mathcal{I}}^j)\boldsymbol{I}_{D_{\mathcal{O}}}$. Hyperparameters λ_1, λ_2 are regularization terms chosen as to constrain the magnitude of the predicted mean and covariance, respectively. The kernel treatment implicit in (1)-(2) assumes the previous choice of a kernel function that depends on the characteristics of the training data. We here consider the squared-exponential kernel

$$k(\boldsymbol{\xi}_{\mathcal{I}}^i, \boldsymbol{\xi}_{\mathcal{I}}^j) = \sigma_f^2 \exp\left(-\frac{1}{l} ||\boldsymbol{\xi}_{\mathcal{I}}^i - \boldsymbol{\xi}_{\mathcal{I}}^j||^2 \right), \quad (4)$$

a common choice in the literature. We hence have that KMP with kernel (4) requires the definition of four hyperparameters $\{\lambda_1, \lambda_2, l, \sigma_f^2\}$. Note the similarity between predictions (1)-(2) and other kernel-based techniques (e.g. GPR, HGP). The main difference is that in KMP the *noise model* is learned through $\boldsymbol{\Sigma}$ which describes both the variability and correlations present in the data throughout the trajectory. This makes KMP a richer representation when compared to GPR or HGP, which assume either constant noise $\hat{\boldsymbol{\Sigma}}_i = \sigma_\epsilon^2 \boldsymbol{I}_{D_{\mathcal{O}}}, \forall i = 1, \dots, N$ (GPR) or input-dependent uncorrelated noise $\hat{\boldsymbol{\Sigma}}_i = \sigma_\epsilon^2(\boldsymbol{\xi}_{\mathcal{I}}^i)\boldsymbol{I}_{D_{\mathcal{O}}}$ (HGP).

IV. UNCERTAINTY-AWARE IMITATION LEARNING WITH KMP

We now demonstrate that KMP provides an estimation of uncertainty through (2), by defaulting to a diagonal matrix completely specified by its hyperparameters in the absence of datapoints (Section IV-A). In addition we propose a control framework to convert the predictions into optimal robot actions (Section IV-B) and the fusion of optimal controllers (Section IV-C).

A. Uncertainty predictions with KMP

In the light of the kernel treatment (2) and the exponential kernel (4), both covariance and uncertainty predictions emerge naturally in the KMP formulation. While the former occur within the training region, the latter arise when querying the model away from the original data.

Lemma 1: The squared exponential kernel (4) goes to zero as $||\boldsymbol{\xi}_{\mathcal{I}}^n - \boldsymbol{\xi}_{\mathcal{I}}^*|| \to +\infty, \forall n = 1, \dots, N$.

Proof: Let us consider $d = ||\boldsymbol{\xi}_{\mathcal{I}}^{\hat{n}} - \boldsymbol{\xi}_{\mathcal{I}}^*||$, where $\hat{n} = \arg\min_n ||\boldsymbol{\xi}_{\mathcal{I}}^n - \boldsymbol{\xi}_{\mathcal{I}}^*||$ is the index of the training point with the minimum distance to the test point $\boldsymbol{\xi}_{\mathcal{I}}^*$.

$$\lim_{d \to +\infty} k(\boldsymbol{\xi}_{\mathcal{I}}^{\hat{n}}, \boldsymbol{\xi}_{\mathcal{I}}^*) = \lim_{d \to +\infty} \sigma_f^2 \exp(-\frac{1}{l} d^2) = 0. \quad (5)$$

∎

Lemma 1 extends to other popular exponential kernels, including the Matérn kernel [8].

Theorem 1: Covariance predictions (2) converge to a diagonal matrix completely specified by the KMP hyperparameters as test inputs $\boldsymbol{\xi}_{\mathcal{I}}^*$ move away from the training dataset, i.e. $d \to +\infty$. Particularly,

$$\lim_{d \to +\infty} \boldsymbol{\Sigma}_{\mathcal{O}}^* = \sigma_f^2 \frac{N}{\lambda_2} \boldsymbol{I}_{D_{\mathcal{O}}}. \quad (6)$$

Proof: Following from Lemma 1 and knowing that $\boldsymbol{k}^* = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^*, \boldsymbol{\xi}_{\mathcal{I}}^1) \dots \boldsymbol{k}(\boldsymbol{\xi}_{\mathcal{I}}^*, \boldsymbol{\xi}_{\mathcal{I}}^N) \end{bmatrix}$ we have $\lim_{d \to +\infty} \boldsymbol{k}^* = \boldsymbol{0}_{D_{\mathcal{O}} \times N D_{\mathcal{O}}}$. Hence

$$\lim_{d \to +\infty} \boldsymbol{\Sigma}_{\mathcal{O}}^* = \lim_{d \to +\infty} \frac{N}{\lambda_2} \boldsymbol{k}^{**}. \quad (7)$$

Moreover we have

$$\boldsymbol{k}^{**} = \boldsymbol{k}\left(\boldsymbol{\xi}_{\mathcal{I}}^*, \boldsymbol{\xi}_{\mathcal{I}}^*\right) = \sigma_f^2 \exp\left(-\frac{1}{l}0\right) = \sigma_f^2 \boldsymbol{I}_{D_{\mathcal{O}}},$$

which replaced in (7) yields (6). ∎

Equation (6) plays a crucial role in our approach. It provides a principled way to know when the model is being queried in regions where data was not present during training. We leverage this information to 1) make the robot compliant when unsure about its actions and 2) let the robot know when to execute control actions pertaining to different KMPs. Moreover, through the dependence on σ_f^2, N and λ_2, one can adjust the expression of uncertainty provided by the model, through the tuning of any of those hyperparameters. For instance, increasing the length of the initialized trajectory distribution N has the effect of scaling the uncertainty. GPR offers a similar property, where the variance prediction converges to the scalar σ_f^2. However this is rather limiting as tuning this hyperparameter can have undesired effects on the mean prediction. In KMP, N and λ_2 do not affect the mean prediction as they do not parameterize the kernel function. Moreover, (2) is typically robust to their choice, providing freedom for tuning while yielding proper predictions (see [5] for details).

B. Computing optimal controllers from KMP

We now propose to use $\boldsymbol{\Sigma}_{\mathcal{O}}^*$ to obtain variable control gains that result in a compliant robot both when the variability and uncertainty are high[1]. We follow the concept introduced in [9] and formulate the problem as a LQR. Let us consider linear systems $\dot{\boldsymbol{\zeta}}_t = \boldsymbol{A}\boldsymbol{\zeta}_t + \boldsymbol{B}\boldsymbol{u}_t$, where $\boldsymbol{\zeta}_t, \dot{\boldsymbol{\zeta}}_t \in \mathbb{R}^{N_S}$ denote the system state at time t and its first-order derivative (N_S is the dimension of the state) and $\boldsymbol{u}_t \in \mathbb{R}^{N_C}$ is a control command, where N_C denotes the number of controlled degrees of freedom. Moreover, $\boldsymbol{A} \in \mathbb{R}^{N_S \times N_S}$ and $\boldsymbol{B} \in \mathbb{R}^{N_S \times N_C}$ represent the state and input matrices. We will stick to task space control and hence make a simplifying assumption, in line with [3], that the end-effector can be modeled as a unitary mass, yielding a double integrator system

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}, \quad (8)$$

[1]In the context of movement synthesis, new inputs occur at every new time step thus we will replace $*$ by t from now on in the notation.

Algorithm 1 Uncertainty-aware imitation learning

Initialization
1: Identify number of sub-tasks P
2: Collect demonstrations $\{\{\{\boldsymbol{\xi}_{\mathcal{I}}^{t,h,p}, \boldsymbol{\xi}_{\mathcal{O}}^{t,h,p}\}_{t=1}^{T}\}_{h=1}^{H}\}_{p=1}^{P}$
3: Generate trajectory distributions $\{\{\boldsymbol{\xi}_{\mathcal{I}}^{n,p}, \hat{\boldsymbol{\mu}}_{n,p}, \hat{\boldsymbol{\Sigma}}_{n,p}\}_{n=1}^{N}\}_{p=1}^{P}$
4: Select hyperparameters $\{\sigma_{f,p}^2, l_p, \lambda_{1,p}, \lambda_{2,p}\}_{p=1}^{P}$ and $\boldsymbol{R}_p$
Movement synthesis
1: *Input:* Test point $\boldsymbol{\xi}_{\mathcal{I}}^{t}$
2: **for** $p = 1, \ldots, P$ **do**
3:     Compute $\boldsymbol{\mu}_{\mathcal{O}}^{t,p}, \boldsymbol{\Sigma}_{\mathcal{O}}^{t,p}$, per (1), (2)
4:     Set $\hat{\boldsymbol{\zeta}}_t^p = \boldsymbol{\mu}_{\mathcal{O}}^{t,p}$ and $\boldsymbol{Q}_t^p = (\boldsymbol{\Sigma}_{\mathcal{O}}^{t,p})^{-1}$
5:     Find optimal gains $\boldsymbol{K}_{t,p}^{\mathcal{P}}, \boldsymbol{K}_{t,p}^{\mathcal{V}}$ and compute $\boldsymbol{u}_t^p$ per (13)
6:     Set $\boldsymbol{\Gamma}_t^p = (\boldsymbol{\Sigma}_{\mathcal{O}}^{t,p})^{-1}$
7: **end for**
8: Compute $\hat{\boldsymbol{u}}_t$ from (12)
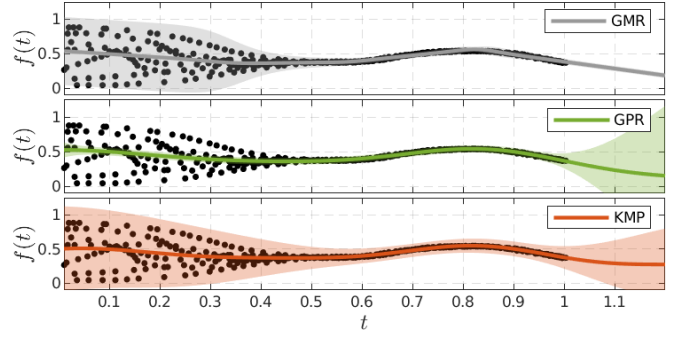9: *Output:* Control command $\hat{\boldsymbol{u}}_t$

Fig. 2: Comparison between KMP, GMR and GPR. Datapoints are plotted in black, solid lines represent the mean and shaded areas correspond to two standard deviations (computed from respective variance estimations).

where $\boldsymbol{0}$ and $\boldsymbol{I}$ are zero and identity matrices of appropriate dimension. We define the end-effector state at t as its Cartesian position and velocity $\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t$, i.e. $\boldsymbol{\zeta}_t = [\boldsymbol{x}_t^{\top} \ \dot{\boldsymbol{x}}_t^{\top}]^{\top}$, and therefore $\boldsymbol{u}_t$ corresponds to acceleration commands.

At every time step t of a task, a KMP is queried with an input test point $\boldsymbol{\xi}_{\mathcal{I}}^{t}$, predicting a mean $\boldsymbol{\mu}_{\mathcal{O}}^{t}$ and a covariance matrix $\boldsymbol{\Sigma}_{\mathcal{O}}^{t}$. We define $\hat{\boldsymbol{\zeta}}_t = \boldsymbol{\mu}_{\mathcal{O}}^{t}$, i.e. the desired state for the end-effector is given by the mean prediction of KMP. For time-driven tasks, where $\boldsymbol{\xi}_{\mathcal{I}}^{t} = t$, a sequence of reference states $\hat{\boldsymbol{\zeta}}_{t=1,\ldots,T}$ can be easily computed and an optimal control command $\boldsymbol{u}_t$ can be found, minimizing

$$c(t) = \sum_{t=1}^{T}(\hat{\boldsymbol{\zeta}}_t - \boldsymbol{\zeta}_t)^{\top}\boldsymbol{Q}_t(\hat{\boldsymbol{\zeta}}_t - \boldsymbol{\zeta}_t) + \boldsymbol{u}_t^{\top}\boldsymbol{R}_t\boldsymbol{u}_t, \quad (9)$$

where $\boldsymbol{Q}_t$ is a $N_S \times N_S$ positive semi-definite matrix that determines how much the optimization penalizes deviations from $\hat{\boldsymbol{\zeta}}_t$ and $\boldsymbol{R}_t$ is an $N_C \times N_C$ positive-definite matrix that penalizes the magnitude of the control commands. Equation (9) is the cost function of the finite horizon LQR and its solution is obtained through backward integration of the Riccati equations (see [3]). In non-time-driven tasks, e.g. when $\boldsymbol{\xi}_{\mathcal{I}}^{t}$ is the state of a human that collaborates with the robot, it is not straightforward to predict a sequence of desired states. In these cases, we resort to the infinite horizon formulation

$$c(t) = \sum_{n=t}^{\infty}(\hat{\boldsymbol{\zeta}}_t - \boldsymbol{\zeta}_n)^{\top}\boldsymbol{Q}_t(\hat{\boldsymbol{\zeta}}_t - \boldsymbol{\zeta}_n) + \boldsymbol{u}_n^{\top}\boldsymbol{R}_t\boldsymbol{u}_n. \quad (10)$$

which is solved iteratively using the algebraic Riccati equation. In both cases (9), (10), the solution is given by a linear state feedback control law

$$\boldsymbol{u}_t = [\boldsymbol{K}_t^{\mathcal{P}} \ \boldsymbol{K}_t^{\mathcal{V}}](\hat{\boldsymbol{\zeta}}_t - \boldsymbol{\zeta}_t) \quad (11)$$

where $\boldsymbol{K}_t^{\mathcal{P}}, \boldsymbol{K}_t^{\mathcal{V}}$ are stiffness and damping gain matrices that drive the system to the desired state.

Finally, we set $\boldsymbol{Q}_t = (\boldsymbol{\Sigma}_{\mathcal{O}}^{t})^{-1}$. Unlike in previous works where a similar choice is made [3], [7], [9], in our approach the unique properties of KMP endow the robot with the ability to modulate its behavior in the face of two different conditions as a consequence of this setting. First, when the

KMP is queried within the training region, full covariance matrices encoding variability and correlations in the demonstrations are estimated, resulting in control gains that reflect the structure in the data. The robot is hence more precise where variability is low (higher gains) and responds to perturbations according to the observed correlations. Second, as the test input deviates from the training data, the robot becomes increasingly more compliant, as a consequence of (6). Intuitively, this makes sense as the robot should be safe when the uncertainty about is actions increases, which is achieved in our formulation by an automatic decrease of the control gains. Our approach is hence the first to permit the robot to be optimal in the region where demonstrations were provided, and safe where data is absent.

C. Fusing optimal controllers

It is often convenient to split the demonstration of complex tasks into smaller, less complex sub-tasks (e.g. grasping a tool and avoiding obstacles). Here we adapt the previously introduced notion of *fusion of controllers* [6] to account for optimal controllers, such as those described in Section IV-B. Let us consider $p = 1, \ldots, P$ candidate controllers generating commands $\boldsymbol{u}_t^p$ that may act on the robot at every time step t (we omit the subscript t in the remainder of this section). In a fusion of controllers, an optimal command is computed as

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \sum_{p=1}^{P}(\boldsymbol{u} - \boldsymbol{u}^p)^{\top}\boldsymbol{\Gamma}^p(\boldsymbol{u} - \boldsymbol{u}^p), \quad (12)$$

where $\boldsymbol{\Gamma}^p$ are weight matrices that regulate the contribution of each individual controller. Examples of $\boldsymbol{\Gamma}^p$ found in the literature include scalar terms that maximize external rewards [20] and precision matrices, either computed from covariance [6], [19] or uncertainty [7]. Equation (12) has an analytical solution given by $\hat{\boldsymbol{u}} = \hat{\boldsymbol{\Sigma}}_u \sum_{p=1}^{P}\boldsymbol{\Gamma}^p\boldsymbol{u}^p$, where $\hat{\boldsymbol{\Sigma}}_u = \left(\sum_{p=1}^{P}\boldsymbol{\Gamma}^p\right)^{-1}$. When $\boldsymbol{u}^p$ and $\boldsymbol{\Gamma}^p$ are viewed as the mean and precision matrix of a Gaussian distribution, this solution corresponds to a Gaussian product [6].

We here propose to use

$$\boldsymbol{u}^p = [\boldsymbol{K}_{t,p}^{\mathcal{P}} \ \boldsymbol{K}_{t,p}^{\mathcal{V}}](\hat{\boldsymbol{\zeta}}_t^p - \boldsymbol{\zeta}_t^p), \quad (13)$$

(a) Examples of handover starting positions.

(b) Handover end position.

Fig. 3: Handover demonstrations. The robot, starting from different initial positions, is moved kinesthetically towards a handover location, where the human hands it a paint roller.

where $[\boldsymbol{K}_{t,p}^{\mathcal{P}} \ \boldsymbol{K}_{t,p}^{\mathcal{V}}]$ are optimal gains estimated using LQR, given the KMP of controller p, and

$$\boldsymbol{\Gamma}^p = (\boldsymbol{\Sigma}_{\mathcal{O}}^p)^{-1} . \qquad (14)$$

As a consequence of (14), controllers with high uncertainty will have negligible influence in the resulting command computed from (12). This permits the demonstration of a task into sub-tasks, whose activation during reproduction depends on their individual uncertainties. Algorithm 1 summarizes the complete approach.

V. EXPERIMENTAL RESULTS

In this section we validate our approach from Section IV using a toy example with synthetic data (Section V-A) and a robot-assisted painting task (Sections V-B and V-C). While we have exploited the latter scenario in previous work [6], here we expand it by considering optimal controllers. The complete task is divided into two sub-tasks: a handover of a paint roller and the application of painting strokes by the robot on a wooden board. In both sub-tasks, the robot motion is driven by the position of the human hand. A supplementary video showing the obtained results is available at http://joaosilverio.weebly.com/uncert.html

A. 1-D regression example with synthetic data

We first consider the regression of a scalar function. Using an artificially generated dataset we trained a KMP with $K = 4$ (number of Gaussian components used in the initialization GMM), $\sigma_f^2 = 1.0$, $l = 1 \times 10^{-2}$, $\lambda_1 = 5$ and $\lambda_2 = 750$. We sampled a trajectory distribution to initialize the KMP with $N = 750$ datapoints. Figure 2 shows the original dataset and the approximated function using KMP, GMR[2] and GPR[3]. While the three techniques accurately predict the mean trend in the function, the variance prediction given by KMP unifies the predictions from GMR and GPR, approximating the variability of the former and the uncertainty of the latter in the appropriate regions of the input space.

B. Robot Handover

We now show that the proposed approach makes the robot track its reference trajectory using optimal gains near the demonstrations while rendering it compliant when the human

[2]computed from the GMM that initialized the KMP
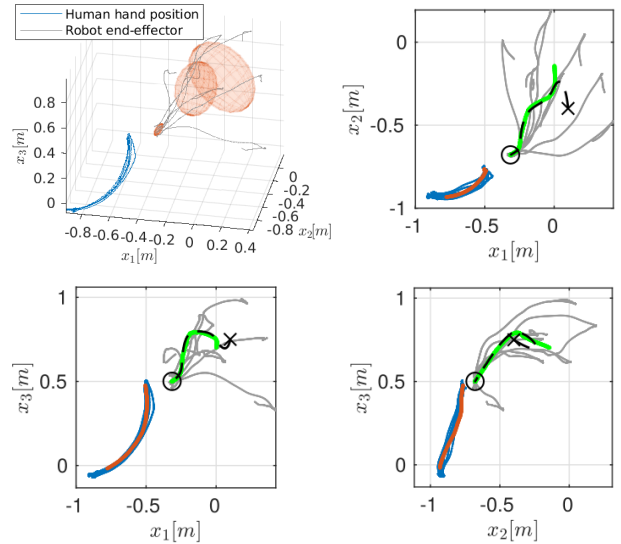[3]with hyperparameters $l = 1 \times 10^{-2}$, $\sigma_\epsilon^2 = 10^{-2}$

Fig. 4: Training data, KMP initialization model and test data. **Top left:** Demonstrated human hand (blue) and robot end-effector (gray) positions. Red ellipsoids show the 3-component GMM used to initialize KMP. **Top right and bottom:** Test human hand (orange), KMP generated (green) and robot measured trajectories (black). '×' and '○' mark start and end of trajectory.

is far from the training data. *The handover of the paint roller is achieved by demonstrating to the robot the location of its end-effector as a function of the human hand position.* Note that object handovers are an extensively studied problem in human-robot collaboration and here we simplify the problem to better focus on showcasing our approach. The human hand and robot end-effector positions are here denoted as $\boldsymbol{x}_H \in \mathbb{R}^3$ and $\boldsymbol{x}_R \in \mathbb{R}^3$ respectively and we wish to learn the mapping $\boldsymbol{x}_H \to \boldsymbol{x}_R$, hence we set $\boldsymbol{\xi}_{\mathcal{I}} = \boldsymbol{x}_H$, $\boldsymbol{\xi}_{\mathcal{O}} = \boldsymbol{x}_R$. We use a KMP with $K = 3$, $\sigma_f^2 = 1.0$, $l = 0.1$, $\lambda_1 = 0.1$ and $\lambda_2 = 1$. Moreover, the KMP is initialized with a trajectory distribution of $N = 500$ points, obtained using GMR at inputs sampled from the GMM. The cost function of the LQR problem is parameterized with $\boldsymbol{R} = 10^{-2}\boldsymbol{I}_{3 \times 3}$ and we follow the infinite horizon formulation minimizing (10), since the input is the human hand position (i.e. not a time-driven motion).

Figure 4 shows the training dataset obtained from 7 demonstrations and the resulting GMM used to initialize the KMP (top-left). Moreover it shows the robot end-effector motion computed for a new human hand trajectory used
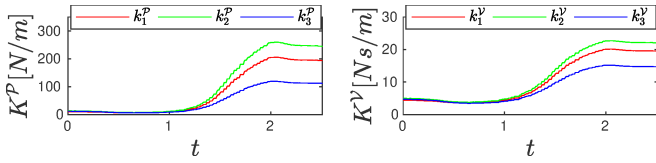
Fig. 5: Stiffness and damping gains during one handover. The gains increase towards the end of the task since the end-effector variability decreases as the robot approaches the handover location.
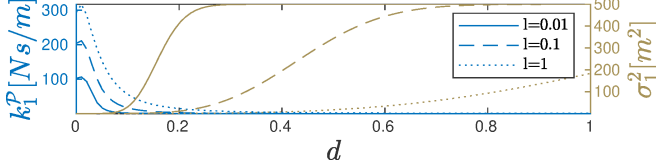

Fig. 6: Stiffness gains (blue) and variance (light brown), plotted for the first task space dimension, as a function of the distance to demonstrations d and different l. Control gains decrease as the distance increases, making the robot gradually more compliant, hence safer, when it does not know what to do.

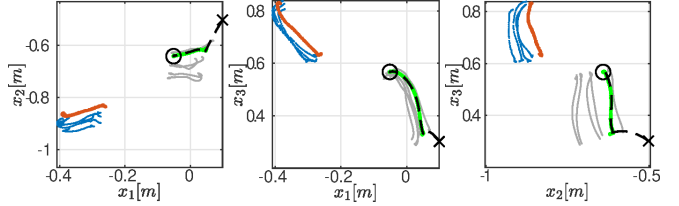Fig. 7: Human operators teach the robot how to apply painting strokes.

Fig. 8: Painting demonstration dataset and reproduction. Training human data (blue), robot data (gray), test human data (orange) and robot desired and observed trajectories (green and black, respectively).

as a test set. As demonstrated, the robot starts at a given position in its task space and moves smoothly towards the handover position, with the learned optimal gains. Figure 5 shows the stiffness and damping gains during one execution, plotted as a function of time. The control gains gradually increase as the human hand approaches the robot, ensuring an accurate tracking of the handover position. This goes in opposite direction to the data covariance that starts large and gradually decreases (Fig. 4 top-left).

Figure 6 shows the estimated gains (left axis) as one moves away from the region where demonstrations were provided. We manually selected one point in the test set and queried the model at several points up to $1m$ away from it along the $+x_1$ direction. In order to facilitate the visualization, we plot one single output dimension and omit the damping gains. Notice the increase in the predicted variance (right axis) as one moves away from the demonstrations, which leads to decreasing control gains. This proves experimentally our proposition in Section IV-A. Moreover, notice the influence of the kernel length scale on how quickly control gains approach 0. Increasing l has the effect of decreasing the distance between points, hence higher values result in a slower increase of uncertainty as one moves away from the data. The squared-exponential nature of the kernel therefore permits regulating the rate at which the robot becomes compliant through the tunning of l. The enclosed video further elucidates the compliance aspect of our approach.

C. Fusion of task space controllers

In addition to the handover of the paint roller, we also teach the robot how to paint. The goal of this experiment is to show that accessing uncertainty, in addition to covariance, permits the fusion of control commands *in a way that different sub-tasks are activated, depending on the state of the human (here defined by its right hand position)*. In this case, the complete painting task was demonstrated partially into two sub-tasks, whose activation will be inferred from

the corresponding models.

We provided 5 demonstrations of painting strokes to the robot as shown in Fig. 7. During these demonstrations, the robot learns to map the wooden board motion (as defined by the human hand) to the movements it should perform with the end-effector. We used the same KMP and LQR parameters as in Section V-B. Figure 8 shows the data used to train the model, together with one reproduction for a human hand trajectory in the neighborhood of the demonstrations. Note that in this case, the differences in covariance in different parts of the end-effector trajectory are not as accentuated as in the handover task. Nonetheless, optimal gains are computed at every moment according to the observed variability and correlations.

Using two KMPs, each one responsible for a sub-task, we reproduced the complete task where the control commands generated by the two candidate controllers were fused as described in IV-C. The complete task took about 2 minutes,
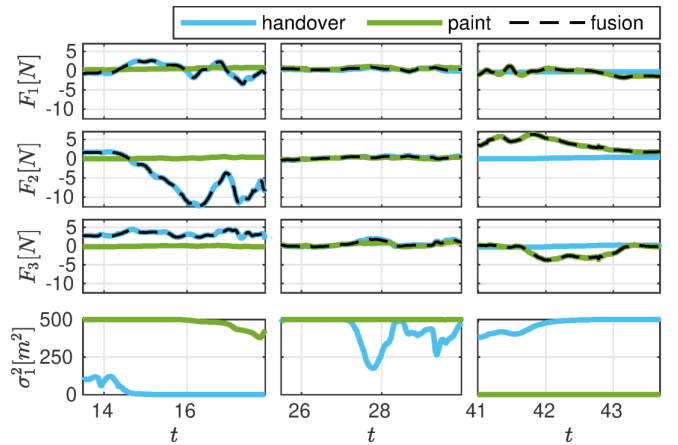

Fig. 9: **Rows 1-3:** Forces generated by each KMP (blue and green) and force used by the robot (black) at three different time intervals of the complete painting task. **Bottom row:** First entry of the covariance matrix (2) predicted by each KMP.

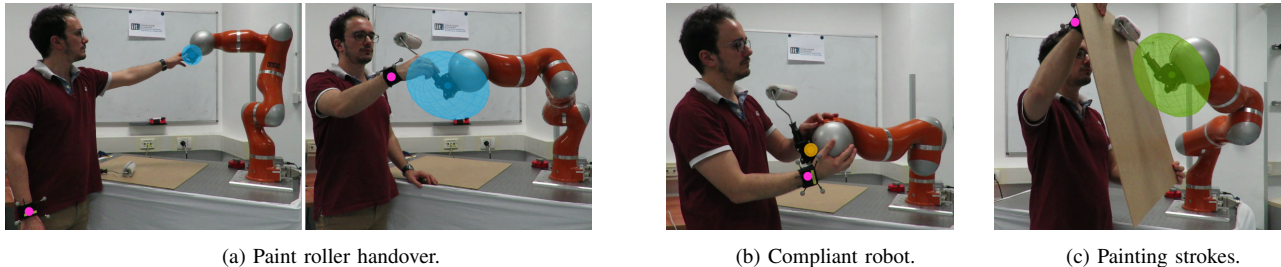(a) Paint roller handover.   (b) Compliant robot.   (c) Painting strokes.

Fig. 10: **Fusion of optimal controllers:** snapshots of the complete painting task. The end-effector stiffness is depicted as an ellipsoid at different moments of the task (larger ellipsoids correspond to higher stiffnesses). The position of the human hand (pink) is used to query two KMPs, whose predictions generated both a reference end-effector position and a covariance matrix from which optimal stiffnesses were computed.

but here we report about $45s$ (the reader is referred to the supplementary video for the full experiment). The first three rows in Fig. 9 show the forces generated by each candidate controller and the force used by the robot. Due to the unit mass assumption in Section IV-B, the acceleration commands are equivalent to desired task space forces $\boldsymbol{F} \in \mathbb{R}^{N_C}$ which are converted into joint space torques through $\boldsymbol{\tau} = \boldsymbol{J}^\top \boldsymbol{F}$ [21] ($\boldsymbol{\tau}$ is a vector of torques and $\boldsymbol{J}$ is the end-effector Jacobian). The bottom row shows the first element of the covariance matrix estimated from (2) (the remainder diagonal elements exhibit similar temporal profiles, hence were omitted). The first column ($\approx 14s - 18s$) corresponds to the beginning of the task, where the paint roller is handed over. The force used by the robot closely matches the one generated by the handover KMP, as its predicted variance is significantly lower during the whole task (bottom plot). Notice the increased value of σ_1^2 around $14s$ – it reflects the high variability in the handover demonstrations at the beginning of the task. This value is consistently below the one generated by the painting KMP. If this were not the case, N could be increased. In the second column of Fig. 9 ($\approx 26s - 30s$) we can see that both KMP generate high variance. This corresponds to a region in between the two sub-tasks hence none of the two should be activated. The fact that the blue line does not reach 500 is due to the human moving slightly closer to the region where the handover was demonstrated. Note however that the observed values are consistently higher than those when tasks are activated. Moreover, notice the low forces during this interval – high covariances yielded minimal control gains resulting in low forces and a compliant robot. Finally in the third column ($\approx 41s - 44s$) we see the task space forces generated during one painting stroke. Notice how, this time, the result from the controller fusion matches the force given by the painting KMP, since the variance for this sub-task is consistently very low (see bottom plot).

Finally, Fig. 10 shows snapshots of different moments of the reproduction. We draw ellipsoids representing full stiffness matrices at the end-effector in the different task moments. These matrices were estimated from the covariance predictions (2) using LQR (Section IV-B). Figure 10(a) shows the two distinct moments of the handover: the beginning, where the end-effector stiffness is low and the user can move the robot around easily, and the end, where the stiffness is high, allowing for the insertion of the paint roller. For an

easier visualization we only plotted the stiffness generated by the *handover KMP* (hence the blue color) since the one from the *painting KMP* was negligible in this part of the task (as we saw in Fig. 9). Figure 10(b) shows a part of the task where none of the two sub-tasks is active. This results in an extremely low stiffness matrix and a fully compliant robot that is safe for the human to interact with and move around in the workspace. In Fig. 10(c) the robot performs a painting stroke on the board, driven by the human hand position, with high stiffness since the demonstrations were consistent in this region. The drawn stiffness ellipsoid resulted from the *painting KMP*, since the one from the *handover KMP* was negligible in the vicinity of this sub-task.

VI. DISCUSSION

In the previous section we showed how KMP can be used to estimate full covariance matrices and uncertainty in order to learn optimal and safe controllers, as well as tasks which are comprised of more than one sub-task. One relevant point of discussion is the fact that, unlike [17], our approach does not explicitly separate between covariance and uncertainty predictions – they are both the result of (2). However, we know a priori the form of the uncertainty predictions, as it is defined by the KMP hyperparameters. If desired, one could potentially assign confidence to a prediction as to whether it corresponds to a covariance matrix or an uncertainty matrix. One way to achieve this could be by resorting to heuristics (e.g. the determinant of the prediction, the Frobenius norm of the distance between matrices) to disambiguate between the two possibilities. Alternatively, one could also exploit the freedom given by the hyperparameters in (6) to accentuate the difference between the two types of prediction. For instance, by setting very low values for λ_2 (unconstraining covariances), one can increase the uncertainty by several orders of magnitude. The same effect would be achieved by sampling more points into the KMP reference trajectory, thus increasing N. This is helped by the fact that, in practice, there are physical limits to how big variability in the data can be (e.g. joint limits, robot workspace size), hence the uncertainty matrix can often be designed so that it is significantly greater than these.

Finally, in the considered setup the two tasks were performed in different parts of the workspace. This was a design choice, as the 3D human hand position was being used to

drive the KMP of each sub-task. However, in practice, our approach can extend to more complicated scenarios. For example, if one was to augment the input vector $\boldsymbol{\xi}_{\mathcal{I}}$ to include other features (e.g. human upper-body configuration, eye gaze), several tasks could potentially overlap in the robot workspace, since there would be more features accounted for by the inputs. This would also lead to a decreased possibility of simultaneously activating undesired sub-tasks.

VII. CONCLUSIONS AND FUTURE WORK

We proposed an imitation learning approach that takes into account the robot uncertainty about its actions, in addition to the variability and correlations in the data, to estimate optimal controllers from demonstrations. The approach was shown to allow for increased safety, as the robot is compliant when uncertain, and efficiency, with the robot using optimal control gains where demonstrations were given. We also showed that demonstrated tasks can be split into sub-tasks that are activated based on their individual uncertainty levels.

In future work we will study how different kernels can be exploited in our framework. Periodic and composite kernels may allow, respectively, for learning tasks that require rhythmic motions or for the usage of different kernels in different parts of a task. Moreover, we plan to exploit the compliance of the robot when uncertain to provide new demonstrations (of the same or new sub-tasks), in an online learning setting. Finally, it should be noted that the prediction capabilities of KMP are independent of the optimal control framework. Predicting full covariance matrices and uncertainty (as shown here), handling start-/via-/end-points [5], multi-dimensional inputs and orientations [22] are distinguishable features of KMP that we aim to leverage in other robotics applications.

REFERENCES

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

[2] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, no. 25, pp. 328–373, 2013.

[3] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Hong Kong, China, May-June 2014, pp. 3339–3344.

[4] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 2616–2624.

[5] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *International Journal of Robotics Research*, vol. 38, no. 7, pp. 833–852, 2019.

[6] 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 Intl Conf. on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October 2018, pp. 6552–6559.

[7] J. Silvério, Y. Huang, L. Rozo, and D. G. Caldwell, "An uncertainty-aware minimal intervention control strategy learned from demonstrations," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October 2018, pp. 6065–6071.

[8] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press, 2006.

[9] J. R. Medina, D. Lee, and S. Hirche, "Risk-sensitive optimal feedback control for haptic assistance," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, May 2012, pp. 1025–1031.

[10] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Seattle, Washington, USA, May 2015, pp. 177–184.

[11] Y. Huang, J. Silvério, and D. G. Caldwell, "Towards minimal intervention control with competing constraints," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October 2018, pp. 733–738.

[12] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *Proc. AAAI Conference on Artificial Intelligence*, 1999, pp. 343–349.

[13] J. R. Medina and S. Hirche, "Uncertainty-dependent optimal control for robot control considering high-order cost statistics," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, September–October 2015, pp. 3995–4002.

[14] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence*, vol. 76, no. 1, pp. 5–23, February 2016.

[15] P. W. Goldberg, C. K. Williams, and C. M. Bishop, "Regression with input-dependent noise: A Gaussian process treatment," in *Advances in Neural Information Processing Systems (NIPS)*, 1998, pp. 493–499.

[16] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian process regression," in *Proceedings of the 24th International Conference on Machine Learning (ICML)*, Oregon, USA, 2007, pp. 393–400.

[17] S. Choi, K. Lee, S. Lim, and S. Oh, "Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Brisbane, Australia, May 2018, pp. 6915–6922.

[18] J. Umlauft, Y. Fanger, and S. Hirche, "Bayesian uncertainty modeling for Programming by Demonstration," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Singapore, May–June 2017, pp. 6428–6434.

[19] S. Calinon and A. G. Billard, "Statistical learning by imitation of competing constraints in joint space and task space," *Advanced Robotics*, vol. 23, no. 15, pp. 2059–2076, 2009.

[20] N. Dehio, R. F. Reinhart, and J. J. Steil, "Multiple task optimization with a mixture of controllers for motion generation," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015, pp. 6416–6421.

[21] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[22] Y. Huang, F. Abu-Dakka, J. Silvério, and D. G. Caldwell, "Generalized orientation learning in robot task space," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019, pp. 2531–2537.