

Gaussian-Process-based Robot Learning from Demonstration

Miguel Arduengo^{1*}, Adrià Colomé¹, Joan Lobo-Prat¹, Luis Sentis² and Carme Torras¹

^{1*}Institut de Robotica i Informatica Industrial (IRI), Barcelona, Spain.

²Human Centered Robotics Laboratory (HCRL), UT Austin, USA.

*Corresponding author(s). E-mail(s): miguel.arduengo@gmail.com;

Abstract

Learning from demonstration allows to encode task constraints from observing the motion executed by a human teacher. We present a Gaussian-Process-based learning from demonstration (LfD) approach that allows robots to learn manipulation skills from demonstrations of a human teacher. By exploiting the potential that Gaussian Process (GP) models offer, we unify in a single, entirely GP-based framework, the main features required for a state-of-the-art LfD approach. We address how GP can be used to effectively learn a policy from trajectories in task space. To achieve an effective generalization across demonstrations, we propose the novel Task Completion Index (TCI) for temporal alignment of task trajectories. Also, our probabilistic GP-based representation allows encoding variability throughout the different phases of the task. Finally, we present a method to efficiently adapt the policy to fulfill new requirements and modulate the robot behavior as a function of task variability. This approach has been successfully tested in a real-world application, namely teaching a TIAGo robot to open different types of doors.

Keywords: Artificial Intelligence, Robot Learning, Gaussian Processes, Learning from Demonstration, Heteroscedasticity

1 Introduction

Robots are progressively spreading to logistic, social and assistive domains. However, in order to become handy co-workers and helpful assistants, they must be endowed with quite different abilities than their industrial ancestors (Torras, 2016).

Moving robots from simple problems to unstructured environments requires a very specific set of skills and knowledge (Billard et al, 2022).

For enabling complex robotics applications, it is much easier for a human to demonstrate the desired behavior rather than attempt to engineer it. This is the main principle behind robot learning from demonstration (LfD). End-users could easily teach robots new tasks without the need of expert programming.

1.1 Learning from Demonstration

Learning from demonstration (LfD) is the paradigm in which robots implicitly learn task constraints and requirements from demonstrations of a human teacher (Ravichandar et al, 2020). This allows more intuitive skill transfer, satisfying a need of opening policy development to non-robotic-experts as robots extend to assistive domains. Flexible models that allow learning the task by extracting relevant motion patterns from the demonstrations, and subsequently apply these patterns to perform the task in different situations, are essential for transferring human skills to robots. Over the last decade, learning from demonstration has been an intensive field of study, for which research interest has done nothing but steadily increase. Also note that, although we use the term learning from demonstration to encompass the field as a whole, other popular terms are used in the literature such as imitation learning, programming by demonstration, and behavioral cloning, among others.

Different learning approaches, namely supervised, reinforcement, and unsupervised, have been used to address a plethora of problems in robot learning. The choice between the different methods is not trivial and depends on the problem of interest (Chen et al, 2020). From a general perspective, to allow robots to learn skills from human demonstrations, we need to develop a system that records demonstrations by experts, learns the ideal behavior from the available demonstrations, and reproduces it.

Several survey papers on robot learning from demonstration provide a distinct overview of the field by answering them from different perspectives (Ravichandar et al, 2020; Osa et al, 2018).

1.2 Trajectory-based robot learning methods

Algorithms that encode skills using trajectory-based representations, are the most dominant family in learning from demonstration research (Colomé and Torras, 2020). These methods rely on low-level controllers to execute the trajectories required to perform the taught skill. Skills are encoded by extracting trajectory patterns from demonstrations (Figure 1), using a variety of techniques to retrieve a generalized shape of the trajectory (Calinon and Lee, 2019). The main reason behind the popularity of these algorithms, is that, assuming that the system is fully actuated (which is the case for most robot manipulators) we do not need any knowledge of the robot dynamics.

For addressing the learning from demonstration problem, we can assume that there exists a direct and learnable function (i.e., the policy) that generates the desired



Fig. 1 The Gaussian-Process-based LfD approach allows to teach robot tasks such as opening doors.

behavior. This policy can be defined as a function that maps available information onto an appropriate action space

$$\pi : \mathcal{X} \longrightarrow \mathcal{Y} \quad (1)$$

where \mathcal{X} represents the inputs required to execute the policy and \mathcal{Y} the action space. The objective is to learn this policy $\pi()$, which allows the reproduction of the skill taught by the expert. For this, the robot is presented with a demonstration (i.e. training) dataset which consists of sample input-action pairs

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N = (X, Y) \quad (2)$$

where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, N stands for the number of samples, and $X \in \mathbb{R}^{\dim(\mathcal{X}) \times N}$ and $Y \in \mathbb{R}^{\dim(\mathcal{Y}) \times N}$ represent the matrices where all the column input and output vectors are aggregated, respectively. From the formulation of the problem, we can see that the first key aspect in LfD involves identifying the appropriate inputs and outputs to the policy. Trajectories are the most popular choice since in a myriad of robotic systems these govern the robot actions.

Another fundamental feature of LfD methods is the possibility of retrieving a probabilistic representation of the policy. This allows a complete description of the task, encoding the uncertainty along with the motion; which is crucial for reflecting the importance of certain points of the task, leading to better generalization capabilities.

Also, in LfD is interesting to adapt the learned motion to unseen scenarios while maintaining the general trajectory shape as in the demonstrations without re-training the model. Commonly, these requirements are expressed as via-point constraints or the blending of multiple movement policies.

In this work, we present a general Gaussian-Process-based learning from demonstration approach. By exploiting the potential that Gaussian Process models offer, we aim to unify in a single, entirely GP-based framework, the main features required for a state-of-the-art LfD approach.

2 State-of-the-art

Over the past two decades, trajectory-based robot learning from demonstration has been intensive field of study. Among the most relevant contributions, the following methods can be highlighted: Dynamic Movement Primitives (DMP) (Ijspeert et al, 2001; Pastor et al, 2009; Saveriano et al, 2021), Probabilistic Movement Primitives (ProMP) (Paraschos et al, 2018; Ewerton et al, 2019; Frank et al, 2021), Gaussian Mixture Model-Gaussian Mixture Regression (GMM-GMR) (Calinon, 2016; Pignat and Calinon, 2019; Pignat et al, 2022), Kernelized Movement Primitives (Huang et al, 2019; Huang et al, 2019), and Gaussian Processes (GP) (Nguyen-Tuong and Peters, 2008; Forte et al, 2010; Schneider and Ertel, 2010). These representations have proved successful at learning and generalizing trajectories. However, each model presents its strengths and shortcomings.

The main advantage of probabilistic-based methods (GMM-GMR, ProMP, KMP and GP) is that they not only retrieve an estimate of the underlying trajectory across multiple demonstrations, but also encode its variability by means of a covariance matrix. This information, which can be inferred from the dispersion of the collected data, can be exploited for the execution of the task, i.e., specifying the robot tracking precision or switching the controller (Silvério et al, 2018).

Unlike probabilistic-based methods, at the cost of not encoding the variability of the task, DMP only requires a single demonstration. Generalization is achieved by assuming trajectories to be solutions of a deterministic dynamical system, achieving remarkable success in generating smooth trajectories from an arbitrary initial state. For capturing higher-order statistics, a unified framework fusing dynamic and probabilistic movement primitives (ProDMPs) (Li et al, 2022), that recovered a linear basis-function representation for the trajectories by solving the dynamical system, has recently been proposed. However, a drawback of DMP, and also ProMP, is that they rely on the manual specification of basis functions, which requires expert knowledge and makes the learning problem with high-dimensional inputs almost intractable. GMM-GMR, in contrast, has proven successful in handling this kind of demonstrations. KMP and GP, by their kernel treatment, can be implemented for manipulation tasks where high-dimensional inputs and outputs are required (Huang et al, 2021).

In LfD is also interesting to transfer the learned motion to unseen scenarios while maintaining the general trajectory shape as in the demonstrations. By exploiting the properties of probability distributions, ProMP, KMP and GP allow for trajectory adaptations with via-points. On the other hand, despite GMM-GMR being formulated in terms of Gaussian distributions, the re-optimization of the learned policy requires to re-estimate the model parameters, which lie in a high-dimensional space. This makes the adaptation process very expensive, which prevents its use in unstructured environments, where the policy adjustment is key.

Besides the generation of adaptive trajectories, another desired property in LfD is extrapolation. In this regard, there is an interesting duality between GMM-GMR and GP representations. The former covariance matrices, model the variability of the trajectories. Conversely, the latter provide a measure of the prediction uncertainty, the variance increasing with the absence of training data. This information is

relevant when trying to generalize the learned motion outside of the demonstrated action space. The simultaneous exploitation of both measures is considered in KMP (Silvério et al, 2019). Moreover, in a recent work, Jaquier et al. (Jaquier et al, 2019) propose a GMM-based GP for encoding the trajectory (GMR-GP), which is a method with enough similarities with KMP, since both are kernel-based. GMR-GP take advantage of the ability of GP to encode prior beliefs through the mean and kernel functions and the capability of GMR to make predictions far from training data. Nevertheless, the improvement with respect to GMR comes at the cost, in both KMP and GMR-GP methods, of an increasing complexity with respect to GP representations. Further, the framework of GP and GMR-GP allows the representation of more complex behaviors that KMP defining a prior for the process.

In the recent years, there has been a growing interest in Gaussian Processes (Schulz et al, 2018). The main advantage of GP over the previously discussed methods, is their ability to encode prior beliefs through the mean and kernel functions. This allows the representation of more complex behaviors in the regions of the action space where demonstration data is sparse. The evaluation of GP models requires however major computational resources with respect to calculation and memory (Nelles, 2020). A few works have studied the use of an entirely GP-based representation in the LfD context (Nguyen-Tuong and Peters, 2008; Forte et al, 2010). Among the most representative is the one presented by Schneider et al. (Schneider and Ertel, 2010). They propose a representation of a pick-and-place task that effectively encodes the task variability using a heteroscedastic GP. Similarly, Umlauf et al. (Umlauf et al, 2017) estimate the prediction uncertainty separately, using Wishart Processes. The learned trajectory is retrieved combining GP and DMP. Neither of these works consider the adaptation of the learned policy. Other works formulate the learning and motion planning problem within a single GP-based framework (Osa et al, 2018; Rana et al, 2017). In these works the entire trajectory is retrieved from an optimization perspective. However, this becomes inefficient as the length of the trajectory and the dimensionality of the learning problem increase. Finally, in a recent work, Wilcox et al. (Wilcox and Yip, 2020) apply GP regression for online non-parametric Bayesian model learning for real-time robot control. However, they do not focus in the trajectory learning problem, but on robot teleoperation.

A drawback of GP is that they are usually only defined in Euclidean space, even though a formulation with non-Euclidean input space is possible in principle (Lang et al, 2018). Thus, when it comes to the modeling of task space trajectories, representation of orientation imposes great challenges, since is accompanied with additional constraints. This is an aspect disregarded in the aforementioned GP-based methods, which is critical in LfD. Some works have successfully addressed this question with DMP (Koutras and Doulgieri, 2019; Abu-Dakka and Kyrki, 2020), GMM-GMR (Zeestraten et al, 2017; Jaquier et al, 2021b) and KMP (Huang et al, 2019; Abu-Dakka et al, 2021). However, in recent works, Lang et al. (Lang et al, 2015; Lang and Hirche, 2017) and Jaquier et al. (Jaquier et al, 2021a) has proposed efficient GP representations for 6-DoF rigid motions. We have adopted in our framework, due to its greater simplicity, the approach developed by Lang et al. (Lang and Hirche, 2017).

Table 1 Comparison between the LfD state-of-the-art and our approach

	DMP	ProMP	GMM-GMR	GP	KMP	GMM-GP	Our approach
<i>High-dimensional Learning</i>	—	—	✓	✓	✓	✓	✓
<i>Via-point Adaptation</i>	—	✓	—	—	✓	✓	✓
<i>Task Variability</i>	—	✓	✓	✓	✓	✓	✓
<i>Prediction Uncertainty</i>	—	—	—	✓	✓	✓	✓
<i>Prior Information</i>	—	—	—	✓	—	✓	✓
<i>Task Space Rotations</i>	✓	✓	✓	—	✓	—	✓

3 Structure and contributions of this paper

In this work, we present a general Gaussian-Process-based learning from demonstration approach. For the purpose of clear comparison, the main contributions of the state-of-the-art and our approach are summarized in Table I.

We show how to achieve an effective representation of the manipulation skill, inferred from the demonstrated trajectories. We unify both, the task variability and the prediction uncertainty, in a single concept we refer to as task uncertainty in the remainder of the paper. Furthermore, in order to achieve an effective generalization across demonstrations, we propose the novel Task Completion Index, for temporal alignment of task trajectories. Finally, we address the adaptation of the policy through via-points, and the modulation of the robot behavior depending on the task uncertainty through variable admittance control.

The paper is structured as follows: in Section 4 we discuss the theoretical aspects of the considered GP models; in Section 5 we present the proposed learning from demonstration framework; in Section 6 we illustrate the main aspects of the paper through a real-world application with the TIAGo robot; finally, in Section 7, we summarize the final conclusions.

4 Gaussian Process Models

In this section we discuss the theoretical background of the proposed LfD approach. First, we present the fundamentals of GP. Then, we address the challenges of modeling rigid-body dynamics with them. Finally, we present how heteroscedastic GP allows to accurately represent the uncertainty of the taught manipulation task.

4.1 Gaussian Process Fundamentals

Intuitively, one can think of a Gaussian process as defining a distribution over functions, and inference taking place directly in the space of functions. Formally, GP are a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams, 2006). It can be completely specified by its mean $m(t)$ and covariance $k(t, t')$ functions:

$$m(t) = \mathbb{E}[f(t)] \quad (3)$$

$$k(t, t') = \mathbb{E}[(f(t) - m(t))(f(t') - m(t'))] \quad (4)$$

where $f(t)$ is the underlying process, $m(t)$ depicts the prior knowledge of its mean, and $k(t, t')$ is symmetric and positive semi-definite (usually referred to as kernel) that must be specified. We are interested in incorporating the knowledge that the training data $\mathcal{D} = \{(t_i, y_i)\}_{i=1}^N$ provides about $f(t)$. We consider that we do not have available direct observations, but only noisy versions y .

Let $\mathbf{m}(t)$ be the vector of the mean function evaluated at all training points t and $K(t, t^*)$ be the matrix of the covariances evaluated at all pairs of training and prediction points t^* . Assuming additive independent identically distributed Gaussian noise with variance σ_n^2 , we can write the joint distribution of the observed target values \mathbf{y} and the function values at the test locations \mathbf{f}^* under the prior as (Nelles, 2020):

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathbf{N} \left(\begin{bmatrix} \mathbf{m}(t) \\ \mathbf{m}(t^*) \end{bmatrix}, \begin{bmatrix} K(t, t) + \sigma_n^2 I & K(t, t^*) \\ K(t^*, t) & K(t^*, t^*) \end{bmatrix} \right) \quad (5)$$

The posterior distribution over functions can be computed by conditioning the joint Gaussian prior distribution on the observations $p(\mathbf{f}^* | t, \mathbf{y}, t^*) \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ where (Nelles, 2020):

$$\boldsymbol{\mu}^* = \mathbf{m}(t^*) + K(t^*, t) [K(t, t) + \sigma_n^2 I]^{-1} [\mathbf{y} - \mathbf{m}(t)] \quad (6)$$

$$\boldsymbol{\Sigma}^* = K(t^*, t^*) - K(t^*, t) [K(t, t) + \sigma_n^2 I]^{-1} K(t, t^*) \quad (7)$$

When we consider only the prediction of one output variable, $k(t, t')$ is a scalar function. The previous concepts can be extended to multiple-output GP (MOGP) by taking a matrix covariance function $\mathbf{k}(t, t')$. Usual approaches to MOGP modelling are mostly formulated around the Linear Model of Coregionalization (LMC) (Alvarez et al, 2012). For a d -dimensional output the kernel is expressed in the following form:

$$\mathbf{B} \otimes \mathbf{k}(t, t') = \begin{bmatrix} B_{11}k_{11}(t_1, t'_1) & \dots & B_{1d}k_{1d}(t_1, t'_d) \\ \vdots & \ddots & \vdots \\ B_{d1}k_{d1}(t_d, t'_1) & \dots & B_{dd}k_{dd}(t_d, t'_d) \end{bmatrix} \quad (8)$$

where $\mathbf{B} \in \mathbb{R}^{d \times d}$ is regarded as the coregionalization matrix and t_i represents the input corresponding to the i -th output. Diagonal elements correspond to the single-output case, while the off-diagonal elements represent the prior assumption on the covariance of two different output dimensions (Liu et al, 2018).

If no a-priori assumption is made, $B_{ij} = 0$ for $i \neq j$ and the MOGP is equivalent to d independent GP. Regarding the form of $k(t, t')$, typically kernel families have free hyperparameters Θ . Such parameters can be determined by maximizing the log marginal likelihood (Rasmussen and Williams, 2006):

$$\log p(\mathbf{y} | t, \Theta) = -\frac{1}{2} \mathbf{y}^T K_y^{-1} \mathbf{y} - \frac{1}{2} \log |K_y| - \frac{N}{2} \log 2\pi \quad (9)$$

where $K_y = K(t, t) + \sigma_n^2 I$. This problem might suffer from local optima.

4.2 Rigid-Body Motion Representation

In the LfD context, representation of trajectories in task space is usually required. However, the modelling of rotations is not straightforward with GP, since the standard formulation is defined for an underlying Euclidean space. A common approach is to use the Euler angles, and exploit that locally the rotation group $SO(3) \simeq \mathbb{R}^3$, allowing distances to be computed as Euclidean. However, when this approximation is no longer valid (e.g. at low sampling frequency or if collected data is sparse) it might lead to inaccurate predictions. To overcome this issue, as proposed in (Lang and Hirche, 2017), from the Euler’s fixed point theorem (Palais and Palais, 2007) rotations can also be parametrized by a set of unit length Euler axes \mathbf{u} together with a rotation angle θ :

$$SO(3) \subset \{ \theta \mathbf{u} \in \mathbb{R}^3 / \|\mathbf{u}\| = 1 \wedge \theta \in [0, \pi] \} \quad (10)$$

This set defines the solid ball $B_\pi(0)$ in \mathbb{R}^3 with radius $0 \leq r \leq \pi$ which is closed, dense and compact. Ambiguity in the representation occurs for $\theta = \pi$. To obtain an isomorphism between the rotation group $SO(3)$ and the axis-angle representation, we fix the axis representation for $\theta = \pi$:

$$\tilde{B}_\pi(0) = B_\pi(0) \setminus \{ \pi \mathbf{u} / u_z < 0 \vee (u_z = 0 \wedge u_y < 0) \vee (u_z = u_y = 0 \wedge u_x < 0) \} \quad (11)$$

where $\mathbf{u} = (u_x, u_y, u_z)$. This parametrization is a minimal and unique $SO(3) \simeq \tilde{B}_\pi(0)$.

Rigid motion dynamics is given by a mapping from time, to translation and rotation $h : \mathbb{R} \rightarrow SE(3)$. Let the translational components be defined by the Euclidean vector $\mathbf{v} \in \mathbb{R}^3$. Then $SE(3)$ is defined isomorphically by $SE(3) \simeq \mathbb{R}^3 \times \tilde{B}_\pi(0)$. Thus, rigid body motion can be represented in MOGP with the 6-dimensional output vector structure $(\mathbf{v}, \theta \mathbf{u}) = (x, y, z, \theta u_x, \theta u_y, \theta u_z)$.

Another possible, more accurate representation, can be achieved with dual quaternions (Lang et al, 2015). However, as shown in (Lang and Hirche, 2017), with the proposed parametrization, a good performance is attained and computations are more efficient.

4.3 Heteroscedastic Gaussian Process

The standard Gaussian Process model assumes a constant noise level. This can be an important limitation when encoding a manipulation task. Consider the example shown in Figure 2: it is evident that while the initial and final positions are highly constrained, that is not the case for the path to follow between such positions. In graphs a) and b) we can see that with a standard approach we accurately represent the mean but not the variability of demonstrations.

Considering an independent normally distributed noise, $\lambda \sim \mathcal{N}(0, r(t))$, where the variance is input-dependent and modeled by $r(t)$. The mean and covariance of the predictive distribution can be modified to (Goldberg et al, 1998):

$$\boldsymbol{\mu}^* = \mathbf{m}(t^*) + K(t^*, t) [K(t, t) + R(t)]^{-1} [\mathbf{y} - \mathbf{m}(t)] \quad (12)$$

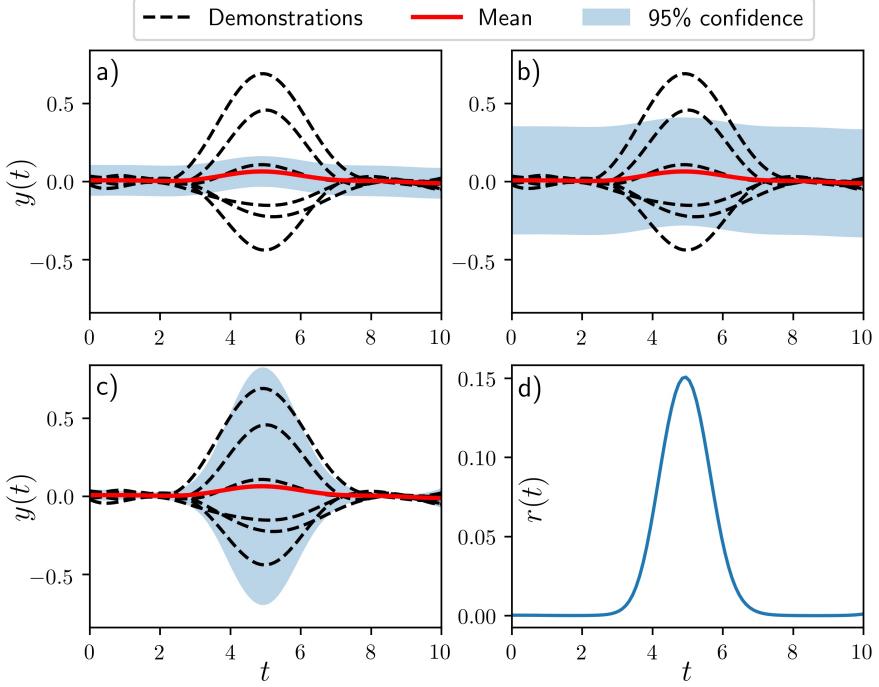


Fig. 2 Standard GP do not accurately model the variability of the demonstrated task as can be seen in a) and b), where it is underestimated and overestimated, respectively. On the other hand, the heteroscedastic GP approach, in c), encodes the variability in the different phases of the task, considering the local noise in d).

$$\Sigma^* = K(t^*, t^*) + R(t^*) - K(t^*, t) [K(t, t) + R(t)]^{-1} K(t, t^*) \quad (13)$$

where $R(t)$ is a diagonal matrix, with elements $r(t)$.

Taking into account the input-dependent noise shown in Figure 2d) the variability in the different phases of the manipulation task is effectively encoded in Figure 2c). This approach is commonly referred to as heteroscedastic Gaussian Process. The main limitation of this method is the trade-off between accuracy in the estimation of the latent noise function, for which more demonstrations are preferred, and the computational complexity of the learning algorithm.

5 Learning from Demonstration Framework

In this section, we present the proposed GP-based LfD framework. First, we formalize the problem of learning manipulation skills from demonstrated trajectories. Then, we propose an approach for encoding the learned policy with GP. Next, we discuss the temporal alignment of demonstrations. We also present a method that allows to adapt the learned policy through via-points. Finally, we study how the uncertainty model of GP can be exploited to stably modulate the robot behavior, varying end-effector virtual dynamics.

5.1 Problem Statement

In LfD we assume that a dataset of demonstrations is available. In the trajectory-learning case, the dataset consists of a set of trajectories \mathbf{s} together with a timestamp $t \in \mathbb{R}$, $\mathcal{D} = \{(t_i, s_i)\}_{i=1}^N$.

Without loss of generality, we will consider $s_i \in SE(3)$. The aim is to learn a policy π that infers, for a given time, the desired end-effector pose s_i^d to perform the taught manipulation task: $s_i^d = \pi(t_i)$. The policy must generate continuous and smooth paths, and generalize over multiple demonstrations.

5.2 Manipulation Task Representation with GP

Representing a manipulation task using heteroscedastic GP models requires the specification of $m(t)$, $k(t, t')$ and $r(t)$. As we have discussed in Section 4.2, a suitable mapping for representing a trajectory is given by the following MOGP:

$$\pi(t) \sim \mathcal{GP}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) : t \longrightarrow (x, y, z, \theta u_x, \theta u_y, \theta u_z) \quad (14)$$

The prior mean function is commonly defined as $m(t) = 0$. Although not necessary in general, if no prior knowledge is available this is a simplifying assumption. The GP covariance function controls the policy function shape. The chosen kernel must generate continuous and smooth paths. Note also that the time parametrization of trajectories is invariant to translations in the time domain. Thus, the covariance function must be stationary. That is, it should be a function of $\tau = t - t'$. The Radial Basis Function (RBF) kernel fulfils all these requirements (Nelles, 2020):

$$k(t, t') = \sigma_f^2 \exp\left(-\frac{[t - t']^2}{2l^2}\right) \quad (15)$$

with hyperparameters l and σ_f .

Moreover, for multidimensional outputs, we have to consider the prior interaction. In the general case, we usually do not have any previous knowledge about how the different components of the demonstrated trajectories relate to each other. Thus, we can assume that the six components are independent a-priori. The matrix covariance function can then be written as (Nelles, 2020):

$$\mathbf{k}(t, t') = \text{diag}\left(\sigma_{f1}^2 e^{([t-t']^2/l_1^2)}, \dots, \sigma_{f6}^2 e^{([t-t']^2/l_6^2)}\right) \quad (16)$$

where $\text{diag}()$ refers to diagonal, and l_i and σ_{fi} correspond to output dimension i .

In Section 4.3 we discussed the convenience of specifying an input-dependent noise function $r(t)$ for encoding the manipulation skill with GP. Usually, it is not known a-priori and must be inferred from the demonstrations. As proposed in (Kersting et al, 2007), first a standard GP can be fit to the data. Its predictions can be used to estimate the input-dependent noise empirically. Then, a second independent GP can be used to model $z(t) = \log[r(t)]$. Let \mathcal{Z} be the set of noise data $\mathbf{z} = \{z_i\}_{i=1}^n$ and its predictions \mathbf{z}^* . The posterior predictive distribution can be approximated by:

$$p(\mathbf{f}^*|D, t^*) = \iint p(\mathbf{f}^*|D, Z, t^*) p(Z|D, t^*) \simeq p(\mathbf{f}^*|D, Z, t^*) \quad (17)$$

where

$$\mathcal{Z} = \arg \max_{z, z^*} p(z, z^*|D, t^*) \quad (18)$$

At this point we have specified all the required functions of the model.

5.3 Temporal Alignment of Demonstrations

For inferring a time dependent policy, the correlation between the temporal and spatial coordinates of two demonstrations of the same task must remain constant. In general, it is very difficult for a human to repeat them at the same velocity. Thus, a time distortion appears (Figure 3a), and should be adequately corrected. Dynamic Time Warping (DTW) (Senin, 2008) is a well-known algorithm for finding the optimal match between two temporal sequences, which may vary in speed.

The algorithm finds a non-linear mapping of the demonstrated trajectories and a reference based on a similarity measure. A common measure in the LfD context is the Euclidean distance. This relies on the assumption that the manipulation task can be performed always following the same path. For instance, consider the case of a pick-and-place task where the objects have to be placed in shelves at different levels (Figure 3).

Using the Euclidean distance as similarity measure will lead to an erroneous temporal alignment (Figure 3b), since intermediate points for placing the object at a higher level can be mapped to ending points of a lower level. We propose to use an index which considers the portion of the trajectory that has been covered for task completion as a similarity measure. We will refer to it as the Task Completion Index (TCI). We define it in discrete form as:

$$\zeta(t_k) = \frac{\sum_{j=1}^k d(\mathbf{s}_j, \mathbf{s}_{j-1})}{\sum_{j=1}^M d(\mathbf{s}_j, \mathbf{s}_{j-1})} \quad \forall k = 1, \dots, M \quad (19)$$

where $\mathbf{s}_j \in SE(3)$ refers to the trajectory point at time instant t_j , $d(\cdot)$ to a scalar distance function and M to the total number of discrete points. Note that $0 = \zeta(t_0) \leq \zeta(t_k) \leq \zeta(t_M) = 1$. As a distance function on $SE(3)$, using the representation discussed in Section 4.2, we define:

$$d(\mathbf{s}_i, \mathbf{s}_j) = \sqrt{\omega_1 [d_{arc}(\theta_i \mathbf{u}_i, \theta_j \mathbf{u}_j)]^2 + \omega_2 \|\mathbf{v}_i - \mathbf{v}_j\|^2} \quad (20)$$

where ω_k are a convex combination of weights for application dependent scaling and $d_{arc}(\cdot)$ is the length of the geodesic between rotations (Lang and Hirche, 2017):

$$d_{arc}(\theta_i \mathbf{u}_i, \theta_j \mathbf{u}_j) = 2 \arccos \left| \cos \frac{\theta_i}{2} \cos \frac{\theta_j}{2} + \sin \frac{\theta_i}{2} \sin \frac{\theta_j}{2} \mathbf{u}_i^T \mathbf{u}_j \right| \quad (21)$$

In Figure 3c we show that the trajectories are warped correctly, allowing then an effective encoding of the manipulation task, with the proposed TCI (Figure 3d).

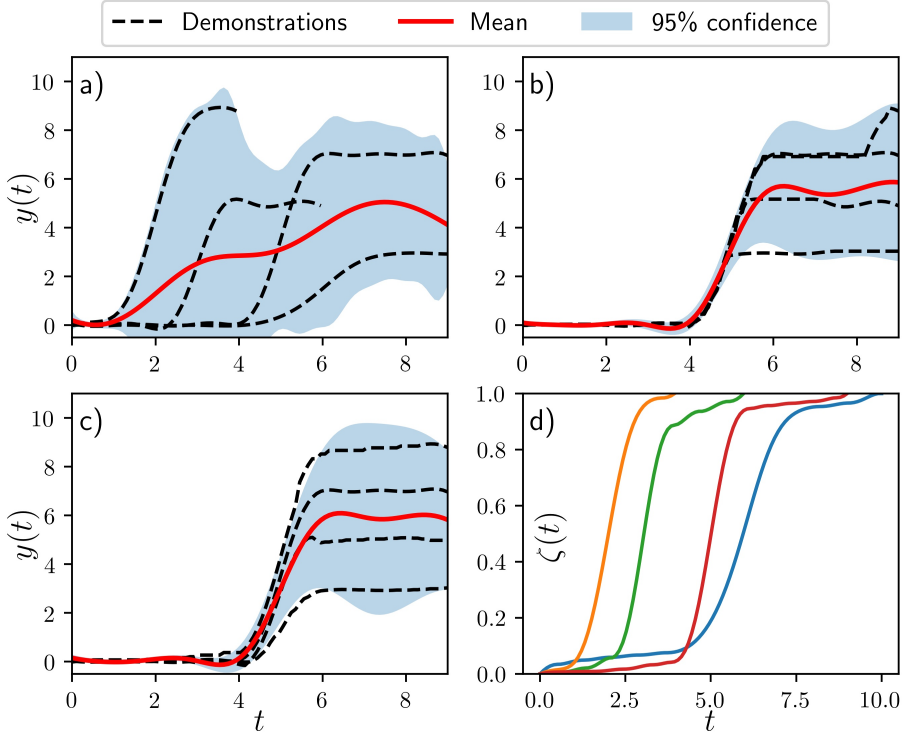


Fig. 3 In a) we observe that due to distortion in time, task constraints are not encoded correctly. In b) the trajectories are aligned with DTW using the Euclidean distance as similarity measure. In c) we show the resulting alignment using the proposed TCI d), as similarity measure.

5.4 Policy Adaptation through Via-points

The modulation of the learned policy through via-points is an important property to adapt to new situations. Let $\mathcal{V} = \{(t_i, s_i^v)\}$ be the set of via-points s_i^v which are desired to be reached by the policy at time instant t_i . In the proposed probabilistic framework, generalization can be implemented by conditioning the policy on both \mathcal{D} and \mathcal{V} . Assuming that the predictive distribution of each set can be computed independently, the conditioned policy is (Deisenroth and Ng, 2015):

$$p(\mathbf{f}^*|D, V, t^*) = p(\mathbf{f}^*|D, t^*) p(\mathbf{f}^*|V, t^*) \quad (22)$$

If $p(\mathbf{f}^*|D, t^*) \sim N(\mu^d, \Sigma^d)$ and $p(\mathbf{f}^*|V, t^*) \sim N(\mu^v, \Sigma^v)$, then, it holds that $p(\mathbf{f}^*|D, V, t^*) \sim N(\mu^{**}, \Sigma^{**})$, where:

$$\mu^{**} = \Sigma^v \left(\Sigma^d + \Sigma^v \right)^{-1} \mu^d + \Sigma^d \left(\Sigma^d + \Sigma^v \right)^{-1} \mu^v \quad (23)$$

$$\Sigma^{**} = \Sigma^d \left(\Sigma^d + \Sigma^v \right)^{-1} \Sigma^v \quad (24)$$

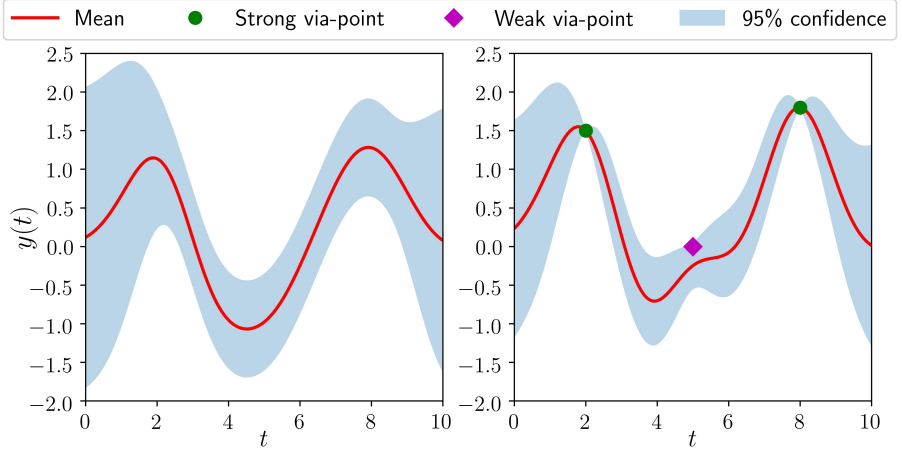


Fig. 4 On the left, a GP model based on the demonstrated trajectories. On the right, the policy adapted through via-points.

The resulting distribution is computed as a product of Gaussians, and is a compromise between the via-point constraints and the demonstrated trajectories, weighted inversely by their variances.

Considering an heteroscedastic GP model for \mathcal{V} (equations 12 and 13), the strength of the via-point constraints can then be easily specified by means of the latent noise function. For instance, via-points with low noise will have a higher relative weight, modifying significantly the learned policy. On the other hand, via-points with a high noise level will produce a more subtle effect. In Figure 4 we illustrate how the distribution adapts to strong and weak defined via-points.

It should be remarked that the posterior predictive distribution of \mathcal{D} only needs to be computed once. Thus, adaptation of the policy just involves a computational cost of $\mathcal{O}(m^3)$, where m is the number of predicted outputs. Since m can be specified, the proposed approach is suitable for on-line applications (for further insight on GP complexity see (Bilj, 2018)).

5.5 Modulation of the Robot Behavior

In LfD is often convenient to adapt the behavior of the robot as a function of the uncertainty in the different phases of the task (Suomalainen et al, 2022). Let the robot end-effector be controlled through a spring-mass-damper model dynamics (Abu-Dakka and Saveriano, 2020):

$$\mathbf{M}(t)\ddot{\mathbf{e}}(t) + \mathbf{D}(t)\dot{\mathbf{e}}(t) + \mathbf{K}_p(t)\mathbf{e}(t) = \mathbf{F}_{\text{ext}}(t) \quad (25)$$

where $\mathbf{M}(t), \mathbf{D}(t), \mathbf{K}_p(t) \in \mathbb{R}^{6 \times 6}$ refer to inertia, damping and stiffness, respectively, and $\mathbf{e}(t) \in \mathbb{R}^{6 \times 1}$ is the tracking error, when subjected to an external force

$\mathbf{F}_{\text{ext}}(t) \in \mathbb{R}^{6 \times 1}$. It can be proved (see (Kronander and Billard, 2016)) that for a constant, symmetric, positive definite \mathbf{M} , and $\mathbf{D}(t)$, $\mathbf{K}_p(t)$ continuously differentiable, the system is globally asymptotically stable if there exists a $\gamma > 0$ such that:

1. $\gamma \mathbf{M} - \mathbf{D}(t)$ is negative semidefinite
2. $\dot{\mathbf{K}}_p(t) + \gamma \dot{\mathbf{D}}(t) - 2\gamma \mathbf{K}_p(t)$ is negative definite

Without loss of generality, we can assume that \mathbf{M} , $\mathbf{D}(t)$, and $\mathbf{K}_p(t)$ are diagonal matrices, since they can always be expressed in a suitable reference frame. Therefore, the system can be uncoupled in six independent scalar systems. Now consider a constant damping ratio δ . Substituting $d(t) = 2\delta\sqrt{mk_p(t)}$ - where m , $d(t)$ and $k_p(t)$ are an arbitrary diagonal element of \mathbf{M} , $\mathbf{D}(t)$ and $\mathbf{K}_p(t)$, respectively - on the second stability condition, it yields the following upper bound for the stiffness derivative:

$$\dot{k}_p(t) < \frac{2\gamma\sqrt{k_p(t)^3}}{\sqrt{k_p(t)} + 2\delta\gamma\sqrt{m}} \quad (26)$$

In order to modulate the robot behavior, we propose the following variable stiffness profile:

$$k_p(t) = k_p^{\max} - \frac{k_p^{\max} - k_p^{\min}}{1 + e^{-\alpha(\sigma(t) - \beta)}} \quad (27)$$

which increases the stiffness inversely to the uncertainty $\sigma(t)$ and saturates at k_p^{\min} and k_p^{\max} for high and low values respectively. Also, note that higher values of the design parameter α give a faster transition between stiff and compliant robot behavior, while β determines a threshold value of $\sigma(t)$ at which the transition starts. Differentiating we have:

$$\dot{k}_p(t) = \alpha k_p(t) \left(1 - \frac{k_p(t)}{k_p^{\max} - k_p^{\min}} \right) \frac{d\sigma(t)}{dt} \quad (28)$$

For a constant $d\sigma(t)/dt$, the maximum value of the stiffness derivative $\dot{k}_p(t)$ is obtained for $k_p(t) = (k_p^{\max} - k_p^{\min})/2$. Thus, substituting in (28), it yields the following upper bound:

$$\dot{k}_p(t) \leq \frac{\alpha}{4} (k_p^{\max} - k_p^{\min}) \frac{d\sigma(t)}{dt} \quad (29)$$

Then, from inspection of the first stability condition, we can see that γ defines a lower bound for the minimum allowed damping $d(t)$.

Given the variable stiffness profile in Equation 27, and assuming constant damping ratio, the most restrictive value is $\gamma = 2\delta\sqrt{k_p^{\min}/m}$. Substituting in (26), we can



Fig. 5 Demonstrations were recorded using an Xsens MVN motion capture system. The teacher opens three doors with different radius.

obtain the following lower bound:

$$\dot{k}_p(t) < \frac{4\delta \sqrt{(k_p^{min})^3}}{(1 + 4\delta^2) \sqrt{m}} \leq \frac{2\gamma \sqrt{k_p(t)^3}}{\sqrt{k_p(t)} + 2\delta \gamma \sqrt{m}} \quad (30)$$

Then, from equations (29) and (30) the following sufficient stability condition can be derived:

$$\frac{d\sigma(t)}{dt} < \frac{16\delta}{\alpha} \frac{\sqrt{(k_p^{min})^3}}{(k_p^{max} - k_p^{min})(1 + 4\delta^2) \sqrt{m}} \quad (31)$$

The control parameters can then be tuned to ensure the satisfaction of this inequality. Note that sharper uncertainty profiles $\sigma(t)$ are more restrictive with respect to variations of the stiffness. For instance, stability is favored by a smaller range $(k_p^{max} - k_p^{min})$ or lower values of α , i.e. slower transition between stiff and compliant behaviors. For the limit cases $k_p^{max} \rightarrow k_p^{min}$ and $\alpha \rightarrow 0$, that is, constant stiffness, stability can be achieved regardless of $\sigma(t)$. It can also be observed, since the right-hand side of the inequality is always positive, that with the proposed variable stiffness profile, stability is ensured if the uncertainty decreases.

6 An Example Application: Door Opening Task

In order to test the proposed GP-based LfD approach, we applied it to the real-world task of opening doors using a TIAGo robot. This is a relevant skill for robots operating in domestic environments (Kim et al, 2004), since they need to open doors when navigating, to pick up objects in fetch-and-carry applications or assist people in their mobility.

6.1 Policy Inference from Human Demonstrations

We performed human demonstrations using an Xsens MVN motion capture system. Right hand trajectories of the human teacher relative to the initial closed door position were recorded for three different doors (Figure 5).

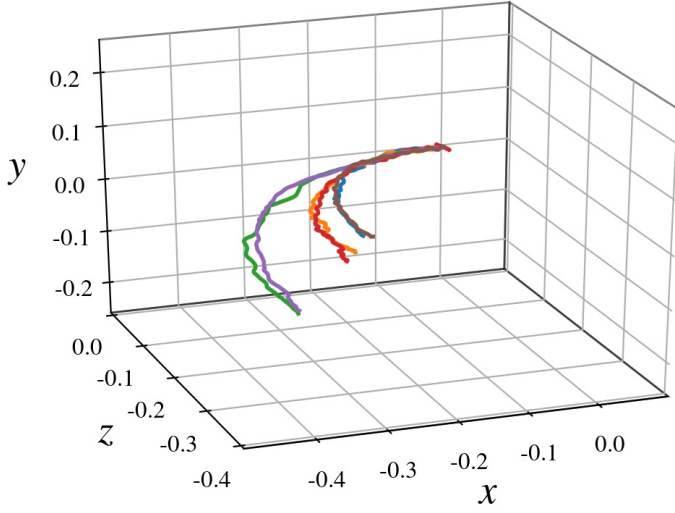


Fig. 6 Right-hand trajectories demonstrations dataset.

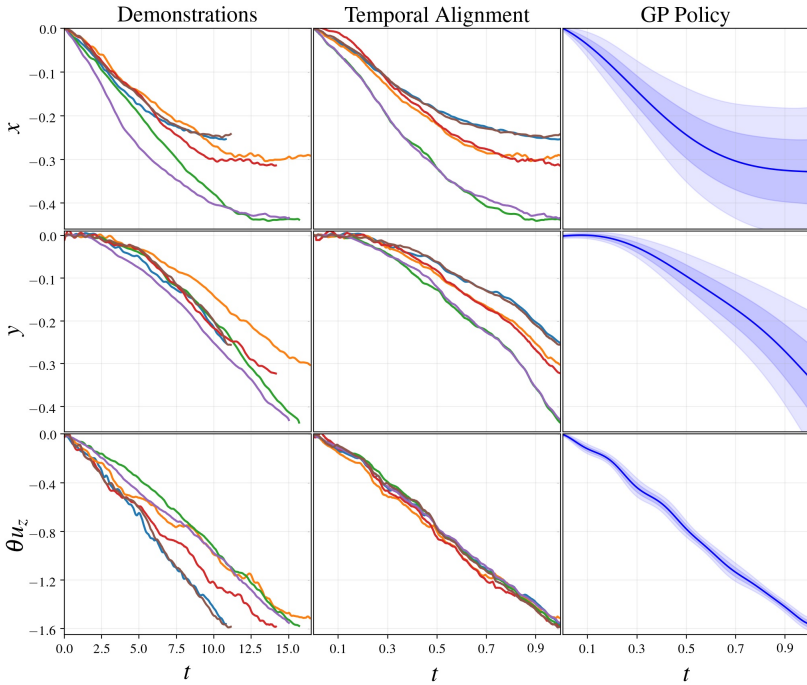


Fig. 7 Inference of the door opening policy from human demonstrations. The outputs are the position, defined by (x, z) , and the orientation defined by θu_z taking the axis-angle representation. On the left column, we have the demonstrations of the door opening motion. On the middle column, the trajectories temporally aligned. On the right column, the inferred Gaussian Process policy, where the dark and light shaded area correspond to the 63% and 95% confidence intervals.

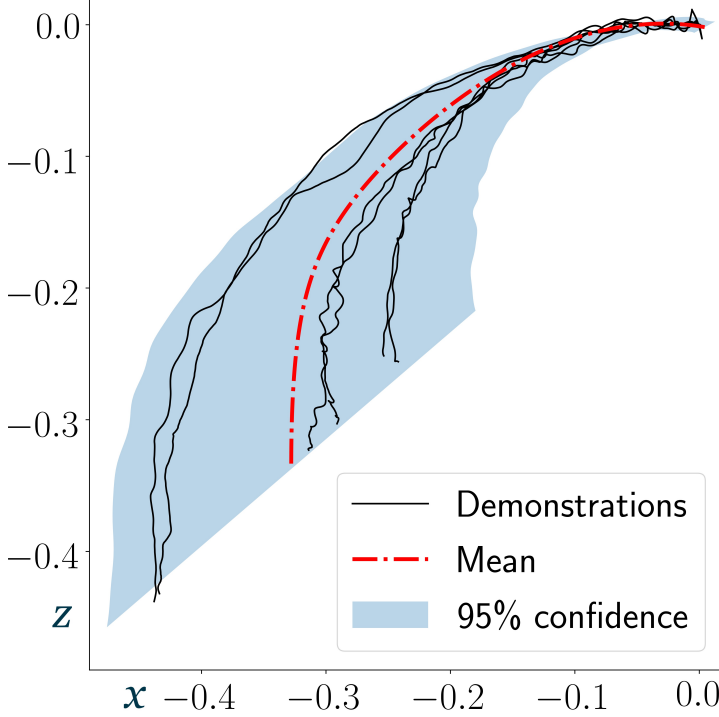


Fig. 8 Door opening policy projected on the $x-z$ plane.

Coordinate axes were chosen such as the pulling direction is parallel to the x axis and the y axis is perpendicular to the floor. The demonstration dataset consisted in a total of 6 trajectories, two per each door (Figure 6).

The main steps of the learning process of the door opening policy are illustrated in Figure 7. The rotation component is encoded using the axis-angle representation. The demonstrated trajectories are aligned with the Dynamic Time Warping algorithm using the task completion index. We can see that the trajectories are warped effectively since they are clearly clustered in three different groups, one for each type of door. Once the trajectories are aligned, we infer the task policy training a heteroscedastic Gaussian Process model on the demonstration data.

We can observe that the model effectively captures the door opening skill. This is more clear in Figure 8, where the task uncertainty has been projected onto the $x-z$ plane. In this case, the variability in the task comes from the uncertainty in the radius of the door, which is reflected in the resulting policy.

6.2 Policy Adaptation and Modulation of the Robot Behavior

During the execution of the task, we can exploit the observations of the motion of the door which is currently being opened to adapt the learned policy. Specifically, we can gather these data by solving the forward kinematics of the robot, and use it to define a set of via-point constraints. By updating this set at each time step we

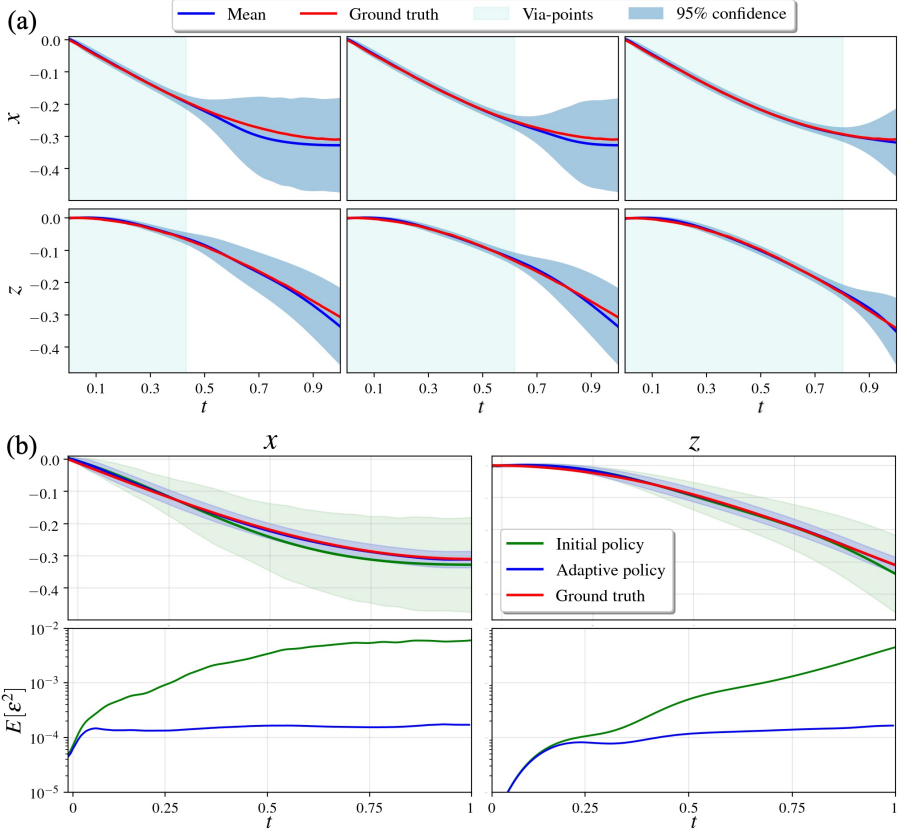


Fig. 9 (a) Evolution of the posterior predictive distribution considering as via-points the observations of the door motion in the light-blue shaded area. (b) The first row shows the comparison between the predictive distribution considering the adaptive policy or the policy based only on human demonstrations. In the second row we can see the mean squared prediction error (32) of each policy.

can adapt the motion online to the current task requirements. In order to evaluate quantitatively the performance of the adaptive policy against the one based solely on the demonstrations, we use the mean squared prediction error (MSPE). Assuming that there exists a ground truth policy $\tilde{\pi}()$, which is the case when opening a door, the MSPE summarizes the predictive ability of the model. Ideally, this value should be close to zero:

$$MSPE = [\epsilon^2] = (E[\pi(t)] - \tilde{\pi}(t))^2 + V[\pi(t)] \quad (32)$$

where $E[\cdot]$ and $V[\cdot]$ refer to the expectancy and the variance, respectively. The evolution of the adaptive policy and the MSPE during the execution of the door opening motion is shown in Figure 9.

We can see that by conditioning on the current observations of the door we are able to reduce the task uncertainty in the near future, converging also the mean to the ground truth. This translates into better performance in terms of the MSPE, as we can see in Figure 9b). It is reduced by almost two orders of magnitude in the final



Fig. 10 TIAGo robot opening the door.

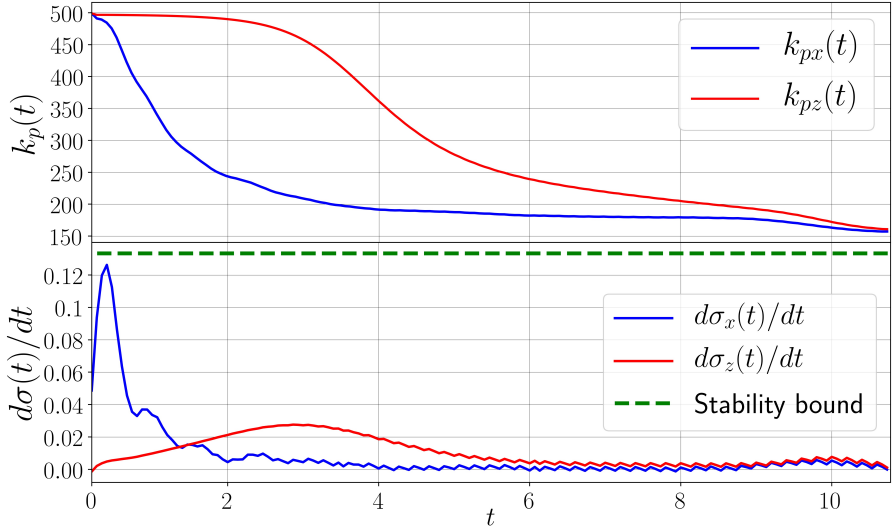


Fig. 11 On top, variable stiffness profile for x and z . Below, the evolution of the uncertainty derivative of the adaptive policy.

stages of the task. With the proposed approach we are able to successfully open the door (Figure 10).

The resulting variable stiffness profile is shown in Figure 11. We have tuned the parameters empirically, being the used values $k_p^{max} = 500$, $k_p^{min} = 100$, $m = 1$, $\delta = 1$, $\alpha = 600$ and $\beta = 0.01$. For simplicity, we have considered the same law for the 6 degrees of freedom. We can observe that the robot behavior is modulated towards a more compliant behavior towards the final phases, where the policy is more uncertain. We can also see that the stability bound is not crossed, which is coherent with the behavior observed in the conducted experiments, where no instabilities occurred.

7 Conclusions

We propose an heteroscedastic multi-output GP policy representation, inferred from demonstrations.

This model considers a suitable parametrization of task space rotations for GP and ensures that only continuous and smooth paths are generated. The introduction of an input-dependent latent noise function allows an effective simultaneous encoding of the prediction uncertainty and the variability of demonstrated trajectories.

In order to establish a correlation between temporal and spatial coordinates, demonstrations must be aligned. We introduce the novel Task Completion Index, a similarity measure that allows to achieve an effective warping when the learned task requires the consideration of different paths.

Adaptation of the policy can be performed by conditioning it on a set of specified via-points. We also introduce a novel computationally efficient method, where the relative importance of the constraints can also be defined. Additionally, we propose an innovative variable stiffness profile that takes advantage of the uncertainty measure provided by the GP model to stably modulate the robot end-effector dynamics.

We applied the proposed learning from demonstration framework to the door opening task and evaluated the performance of the learned policy through real-world experiments with the TIAGo robot. Results show that the manipulation skill is effectively encoded and a successful reproduction can be achieved by taking advantage of the policy adaptation and robot behavior modulation approaches.

In future works we intend to improve the scalability of the learning algorithm by exploiting the structure of replications, and the adaptability of the model by incorporating task variables. This would allow us to apply our method for learning complex robot skills, such as cloth manipulation.

Acknowledgements

This work is partially funded by ERC Advanced Grant H2020-741930 (project CLOTHILDE).

References

- Abu-Dakka FJ, Kyrki V (2020) Geometry-aware dynamic movement primitives. In: IEEE International Conference on Robotics and Automation (ICRA), pp 4421–4426
- Abu-Dakka FJ, Saveriano M (2020) Variable Impedance Control and Learning: A Review. *Frontiers in Robotics and AI* 7
- Abu-Dakka FJ, Huang Y, Silvério J, et al (2021) A probabilistic framework for learning geometry-based robot manipulation skills. *Robotics and Autonomous Systems* 141:103,761
- Alvarez MA, Rosasco L, Lawrence ND (2012) Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning* 4(3)
- Bilj HL (2018) LQG and Gaussian Process Techniques for Fixed-Structure Wind Turbine Control. PhD Dissertation, Delft University of Technology

- Billard A, Mirrazavi S, Figueroa N (2022) Learning for Adaptive and Reactive Robot Control - A Dynamical Systems Approach. The MIT Press
- Calinon S (2016) A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics* 9(1)
- Calinon S, Lee D (2019) Learning Control. In: Vadakkepat P, Goswami A (eds) *Humanoid Robotics: A Reference*. Springer, pp 1261–1312
- Chen J, Xiao Z, Xing H, et al (2020) STDPG: A Spatio-Temporal Deterministic Policy Gradient Agent for Dynamic Routing in SDN. In: *IEEE International Conference on Communications (ICC)*, Dublin, Ireland, pp 1–6
- Colomé A, Torras C (2020) Reinforcement Learning of Bimanual Robot Skills. *Springer Tracts in Advanced Robotics (STAR)*, vol. 134. Springer International Publishing
- Deisenroth M, Ng JW (2015) Distributed Gaussian Processes. In: *32nd International Conference on Machine Learning (ICML)*, pp 1481–1490
- Ewerton M, Arenz O, Maeda G, et al (2019) Learning trajectory distributions for assisted teleoperation and path planning. *Frontiers in Robotics and AI* 6
- Forte D, Ude A, Kos A (2010) Robot learning by Gaussian Process regression. In: *19th Workshop on Robotics in Alpe-Adria-Danube Region*, pp 303–308
- Frank F, Paraschos A, van der Smagt P, et al (2021) Constrained probabilistic movement primitives for robot trajectory adaptation. *IEEE Transactions on Robotics* pp 1–19
- Goldberg P, Williams C, Bishop C (1998) Regression with input-dependent noise: A Gaussian Process treatment. *Advances in Neural Information Processing Systems* 10:493–499
- Huang Y, Abu-Dakka FJ, Silvério J, et al (2019) Generalized orientation learning in robot task space. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp 2531–2537
- Huang Y, Rozo L, Silvério J, et al (2019) Kernelized Movement Primitives. *The International Journal of Robotics Research* 38(7):833–852
- Huang Y, Rozo L, Silvério J, et al (2019) Non-parametric imitation learning of robot motor skills. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp 5266–5272
- Huang Y, Abu-Dakka FJ, Silvério J, et al (2021) Toward orientation learning and adaptation in cartesian space. *IEEE Transactions on Robotics* 37(1):82–98

- Ijspeert AJ, Nakanishi J, Schaal S (2001) Trajectory formation for imitation with non-linear dynamical systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 752–757
- Jaquier N, Ginsbourger D, Calinon S (2019) Learning from demonstration with model-based Gaussian Process. In: 3rd Conference on Robot Learning (CoRL), Osaka, Japan
- Jaquier N, Borovitskiy V, Smolensky A, et al (2021a) Geometry-aware bayesian optimization in robotics using riemannian matérn kernels. In: 5th Conference on Robot Learning (CoRL), London, UK
- Jaquier N, Rozo L, Caldwell DG, et al (2021b) Geometry-aware manipulability learning, tracking, and transfer. *The International Journal of Robotics Research* 40(2-3):624–650
- Kersting K, Plagemann C, Pfaff P, et al (2007) Most-likely Heteroscedastic Gaussian Process regression. In: ACM Proceeding Series, pp 393–400
- Kim D, Kang JH, Hwang CS, et al (2004) Mobile robot for door opening in a house. In: Knowledge-Based Intelligent Information and Engineering Systems (KES), Part II, pp 596–602
- Koutras L, Doulgeri Z (2019) A correct formulation for the Orientation Dynamic Movement Primitives for robot control in the Cartesian space. In: 3rd Conference on Robot Learning (CoRL), Osaka, Japan
- Kronander K, Billard A (2016) Stability considerations for variable impedance control. *IEEE Transactions on Robotics* 32(5):1298–1305
- Lang M, Hirche S (2017) Computationally efficient rigid-body Gaussian Process for motion dynamics. *IEEE Robotics and Automation Letters* 2(3)
- Lang M, Kleinstaub M, Dunkley O, et al (2015) Gaussian Process dynamical models over dual quaternions. In: European Control Conference (ECC), pp 2847–2852
- Lang M, Kleinstaub M, Hirche S (2018) Gaussian Process for 6-DoF rigid motions. *Autonomous Robots* 42(6)
- Li G, Jin Z, Volpp M, et al (2022) ProDMPs: A Unified Perspective on Dynamic and Probabilistic Movement Primitives. *ArXiv abs/2210.01531*
- Liu H, Cai J, Ong YS (2018) Remarks on Multi-Output Gaussian Process Regression. *Knowledge-Based Systems* 144:102–121
- Nelles O (2020) From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes, 2nd edn. Springer Cham

- Nguyen-Tuong D, Peters J (2008) Local Gaussian Process regression for real-time model-based robot control. In: IEEE International Conference on Intelligent Robots and Systems, pp 380–385
- Osa T, Pajarinen J, Neumann G, et al (2018) Foundations and Trends in Robotics
- Osa T, Sugita N, Mitsuishi M (2018) Online trajectory planning and force control for automation of surgical tasks. IEEE Transactions on Automation Science and Engineering 15:675–691
- Palais B, Palais R (2007) Euler’s fixed point theorem: The axis of a rotation. Journal of Fixed Point Theory and Applications 2:215–220
- Paraschos A, Daniel C, Peters J, et al (2018) Using probabilistic movement primitives in robotics. Autonomous Robots 42(3):529–551
- Pastor P, Hoffmann H, Asfour T, et al (2009) Learning and generalization of motor skills by learning from demonstration. In: IEEE International Conference on Robotics and Automation, pp 763–768
- Pignat E, Calinon S (2019) Bayesian Gaussian Mixture Model for robotic policy imitation. IEEE Robotics and Automation Letters (RA-L) 4(4)
- Pignat E, Silvério J, Calinon S (2022) Learning from demonstration using products of experts: Applications to manipulation and task prioritization. The International Journal of Robotics Research 41(2):163–188
- Rana MA, Mukadam M, Ahmadzadeh SR, et al (2017) Towards robust skill generalization: unifying learning from demonstration and motion planning. In: Conference on Robot Learning (CoRL), CA, USA
- Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. The MIT Press
- Ravichandar H, Polydoros A, Chernova S, et al (2020) Recent advances in robot learning from demonstration. Annual Review of Control, Robotics, and Autonomous Systems 3
- Saveriano M, Abu-Dakka FJ, Kramberger A, et al (2021) Dynamic Movement Primitives in Robotics: A Tutorial Survey. ArXiv abs/2102.03861
- Schneider M, Ertel W (2010) Robot learning by demonstration with local Gaussian Process regression. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 255–260
- Schulz E, Speekenbrink M, Krause A (2018) A tutorial on Gaussian Process regression: Modelling, exploring, and exploiting functions. Journal of Mathematical Psychology 85:1–16

- Senin P (2008) Dynamic Time Warping algorithm review. Tech. rep., Computer Science Department, University of Hawaii at Manoa, Honolulu, USA
- Silvério J, Huang Y, Rozo L, et al (2018) Probabilistic learning of torque controllers from kinematic and force constraints. In: IEEE International Conference on Intelligent Robots and Systems, pp 1–8
- Silvério J, Huang Y, Abu-Dakka F, et al (2019) Uncertainty-aware imitation learning using Kernelized Movement Primitives. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp 90–97
- Suomalainen M, Karayiannidis Y, Kyrki V (2022) A survey of robot manipulation in contact. *Robotics and Autonomous Systems* 156:104,224
- Torras C (2016) Service Robots for Citizens of the Future. *European Review* 24(1):17–30
- Umlauft J, Fanger Y, Hirche S (2017) Bayesian uncertainty modeling for programming by demonstration. In: IEEE International Conference on Robotics and Automation (ICRA), pp 6428–6434
- Wilcox B, Yip MC (2020) Sparse Online Locally Adaptive Regression Using Gaussian Processes for Bayesian Robot Model Learning and Control. *IEEE Robotics and Automation Letters* 5(2):2832–2839
- Zeestraten MJ, Havoutis I, Silvério J, et al (2017) An approach for imitation learning on riemannian manifolds. *IEEE Robotics and Automation Letters* 2:1240–1247