Contextual Policy Search for Micro-Data Robot Motion Learning through Covariate Gaussian Process Latent Variable Models

Juan Antonio Delgado-Guerrero[1], Adrià Colomé[1], and Carme Torras[1]

***Abstract*— In the next few years, the amount and variety of context-aware robotic manipulator applications is expected to increase significantly, especially in household environments. In such spaces, thanks to programming by demonstration, non-expert people will be able to teach robots how to perform specific tasks, for which the adaptation to the environment is imperative, for the sake of effectiveness and users safety. These robot motion learning procedures allow the encoding of such tasks by means of parameterized trajectory generators, usually a Movement Primitive (MP) conditioned on contextual variables. However, naively sampled solutions from these MPs are generally suboptimal/inefficient, according to a given reward function. Hence, Policy Search (PS) algorithms leverage the information of the experienced rewards to improve the robot performance over executions, even for new context configurations. Given the complexity of the aforementioned tasks, PS methods face the challenge of exploring in high-dimensional parameter search spaces. In this work, a solution combining Bayesian Optimization, a data-efficient PS algorithm, with covariate Gaussian Process Latent Variable Models, a recent Dimensionality Reduction technique, is presented. It enables reducing dimensionality and exploiting prior demonstrations to converge in few iterations, while also being compliant with context requirements. Thus, contextual variables are considered in the latent search space, from which a surrogate model for the reward function is built. Then, samples are generated in a low-dimensional latent space, and mapped to a context-dependent trajectory. This allows us to drastically reduce the search space with the covariate GPLVM, e.g. from 105 to 2 parameters, plus a few contextual features. Experimentation in two different scenarios proves the data-efficiency and the power of dimensionality reduction of our approach.**

I. Introduction

Generalizing previous learned robot motion knowledge and adapting to multiple context configurations are inherent challenges of the forthcoming robotic applications, particularly in household environments. Robot programming by demonstration through kinesthetic teaching can be useful in such scenarios [1], as it allows lay people to instruct robots different tasks directly, without explicitly programming each detail. Each task can be modelled with a Movement Primitive (MP), i.e., a parameterized generative model which is initially fitted with the user demonstrations. Subsequently, the robot skills can be honed by updating the MP parameters through trial-and-error within the framework of Policy Search (PS), a branch of Reinforcement Learning (RL) [2] responsible for resolving which trajectories to evaluate in consideration of the rewards of each execution. Thus, PS algorithms have proved successful in several robotic applications [3], including the contextual case, in which robots are required to adapt to changing environments [4], [5], [6], [7].

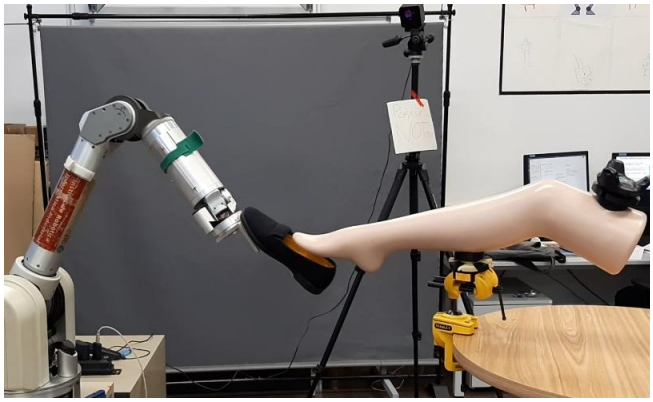

Fig. 1: Barrett's WAM robot inserting a shoe into a mannequin's foot (which changes its position and orientation) after learning a policy with our proposed method.

However, model-free PS algorithms, such as Relative Entropy Policy Search (REPS) [8], can struggle with high-dimensional parameter spaces, requiring of numerous samples until convergence and hence becoming technically unfeasible for some robotic uses. Thus, several approaches exist to overcome this data inefficiency. The so-called Micro-Data PS methods [9] focus on reducing the number of robot trials, by building surrogate models of the robot/environment dynamics, the reward function, or leveraging prior knowledge. These strategies are often combined. For example, PILCO [10] and Black-DROPS [11] are model-based PS methods that use models of both the dynamics and the reward functions. Nevertheless, the majority of the proposed methods in [9] are not suitable for task generalization or do not deal with it, as context variations typically incur additional robot interaction. Moreover, most of the existing contextual PS methods are model-free, such as contextual REPS, a special case of Hierarchical REPS [12], learn models of the robot and its environment, like GPREPS [4], or learn models of the reward function through active learning, occasionaly querying an expert user to rate the robot task performance, as in [13].

This work has been developed in the context of the project CLOTHILDE ("CLOTH manIpulation Learning from DEmonstrations"), which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Advanced Grant agreement No 741930). This work is also supported by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI MDM-2016-0656.

[1]Institut de Robòtica i Informàtica Industrial (IRI), CSIC-UPC, Barcelona, Spain. [jdelgado,acolome,torras]@iri.upc.edu.

In this paper, we assume that tasks at hand can be modelled as MPs, physically executed and evaluated by means of a reward function considered as a black box, and prior information on parameters is available through initial demonstrations, but instead, we cannot model their dynamics. On this basis, the proposed solution continues along the research line presented in [14]. Therefore, to speed up convergence, we build a surrogate model of reward and apply Bayesian Optimization (BO) in the latent space arisen from a Dimensionality Reduction (DR) of the PS parameter space, since BO algorithms do not perform well with high-dimensional search spaces. This enhanced method is implemented in a convenient manner to ensure contextual adaptation, and it significantly increments the DR power and data efficiency with respect to [14]. In particular, Upper Confidence Bound (UCB) [15], [16] and Gaussian Process (GP) regression [17] have been used to learn the model of the expected return, and covariate Gaussian Process Latent Variable Models (c-GPVLM) [18] have been applied for context-aware DR.

Several methods on robot motion learning in latent spaces through DR have proved to be successful [19], [20], [21], [22], including the contextual case [23], but the capacity of linear models for DR is limited for higher-dimensional spaces. Therefore, in this work we applied c-GPLVM, a novel extension of GPLVM [24], which are non-linear.

After this DR is completed, we make use of UCB in the resulting latent space to decide which samples to evaluate, according to the context requirements. Next, we reproject those samples to the high-dimensional space, execute and evaluate them, and then update the surrogate model of the reward function. This last idea has proved to be very useful. For example, in [13], UCB and GP regression are successfully used to learn the model of the expected return from an *outcome* space, although, in that case, those outcomes, which represent relevant features of the trajectories, are assumed to be known. Instead, in our proposal, such relevant features are found through latent variables that encode the trajectories and the context specifications. For this reason, arbitrary outcomes can be sampled and the policy search space is highly reduced.

This paper is organized as follows: Section II briefly introduces the concepts used in the paper, such as Movement Primitives and contextual Policy Search, Gaussian Processes (GP), Gaussian Process Latent Variable Models (GPLVM), covariate Gaussian Process Latent Variable Models (c-GPLVM), Bayesian Optimization (BO) and Upper Confidence Bound (UCB). Section III defines the proposed approach and Section IV presents the results obtained with this method. Section V concludes the paper and proposes future directions.

II. Preliminaries

A. Movement Primitives and contextual Policy Search

MPs are a standard approach to model, encode and learn similar motion trajectories, they being widely used as parameterized trajectory generators in PS. In this paper, linear basis function models with uniformly distributed normalized Gaussian kernels over time have been used for this encoding [25]. Thus, given a number of basis functions per degrees of freedom (DoF), N_f, the position and/or velocity state vector $\mathbf{z}_t$ can be represented as

$$\mathbf{z}_t = \mathbf{\Psi}_t^T \boldsymbol{\omega} + \boldsymbol{\epsilon}_z, \tag{1}$$

where $\mathbf{\Psi}_t^T = I_{N_d} \otimes \mathbf{\Phi}_t^T$, I_{N_d} being the N_d-dimensional identity matrix, with N_d the number of DoFs of the robot, $\mathbf{\Phi}_t$ an N_f-dimensional column vector with the Gaussian kernels associated to one DoF at time t, and $\boldsymbol{\epsilon}_z$ a zero-mean Gaussian noise. Therefore, given a set of demonstration trajectories $\boldsymbol{\tau}_n = \{\mathbf{z}_t^n\}_{t=1..N_t}$, $n = 1..N$, the weights $\boldsymbol{\omega}_n$ of each demonstration can be computed through least squares.

This approach is convenient for robot PS, as it provides a compact representation of complex tasks, and it allows correlated local exploration by varying parameters $\boldsymbol{\omega}_n$, thus generating smooth trajectories. The policy is then defined as the trajectory tracking controller that follows $\boldsymbol{\tau}_n$, whose motion is represented by $\boldsymbol{\omega}_n$. The aim of contextual PS is to learn how to vary $\boldsymbol{\omega}_n$, depending on the context variables $\mathbf{s}_n$.

B. Gaussian Processes

Gaussian Processes (GP) [17] are the generalization of multivariate Gaussian distributions, over finite-dimension vectors, to infinite-dimension, over functions. Otherwise said, a GP $\boldsymbol{f}$ is an infinite-dimension stochastic process such that, for any finite set of indices $x_1, ..., x_n$, the random variables $\boldsymbol{f}(x_1), ..., \boldsymbol{f}(x_n)$ have a joint Gaussian distribution completely defined by its mean function m and covariance function k, which is symmetric and positive semi-definite:

$$\boldsymbol{f}(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{2}$$

Usually, the mean function is chosen to be the zero function, $m(\mathbf{x}) = 0$. On the contrary, many options are available in literature for defining the covariance function k. In this paper, the popular squared exponential kernel combined with a vector of automatic relevance determination has been used for this purpose:

$$k(\mathbf{x_i}, \mathbf{x_j}) = \sigma^2 \exp\left(-\frac{1}{2}(\mathbf{x_i} - \mathbf{x_j})^T \text{diag}(\boldsymbol{\ell})^{-2}(\mathbf{x_i} - \mathbf{x_j})\right), \tag{3}$$

where σ is the kernel variance parameter and $\boldsymbol{\ell}$ is the length-scale vector parameter.

Moreover, regression models can be built from GPs, $\mathbf{y} = \boldsymbol{f}(\mathbf{x}) + \epsilon$, being ϵ a noise Gaussian distribution. Thus, considering a set of N observations in matrix form $\{\mathbf{X}, \mathbf{Y}\}$, with $\mathbf{X} \in \mathbb{R}^{N \times Q}$, $\mathbf{Y} \in \mathbb{R}^{N \times D}$, $\boldsymbol{f}$ can be used to predict the value of $\mathbf{y}_{N+1}$, given $\mathbf{x}_{N+1}$. From the properties of $\boldsymbol{f}$ and the Gaussian identities, the following expressions are derived:

$$\begin{aligned} &P(\mathbf{y}_{N+1}|\mathbf{X}, \mathbf{Y}, \mathbf{x}_{N+1}) = \\ &= \mathcal{N}(\mu_t(\mathbf{x}_{N+1}), \sigma_t^2(\mathbf{x}_{N+1}) + \sigma_{noise}^2), \end{aligned} \tag{4}$$

where

$$\mu_t(\mathbf{x}_{N+1}) = \mathbf{k}^T[\mathbf{K} + \sigma^2_{noise} I_N]^{-1}\mathbf{Y} \quad (5)$$

$$\sigma^2_t(\mathbf{x}_{N+1}) = k(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) - \mathbf{k}^T[\mathbf{K} + \sigma^2_{noise} I_N]^{-1}\mathbf{k} \quad (6)$$

$$\mathbf{K}_{i,j} = k(x_i, x_j) \quad i, j = 1..N \quad (7)$$

$$\mathbf{k}_i = k(x_{N+1}, x_i) \quad i = 1..N \quad (8)$$

C. Gaussian Process Latent Variable Models

Initially conceived for the visualization of high-dimensional spaces [26], GPLVM are a feature extraction method that can be considered as multiple-output GP regression models, outlined in Sec. II-B, built from the output data only. By optimization of certain parameters, these models learn a low-dimensional representation $\mathbf{X} \in \mathbb{R}^{N \times Q}$, from a set of observed data, $\mathbf{Y} \in \mathbb{R}^{N \times D}$, being ideally $Q \ll D$ for the purpose of DR. As a result of the optimization, GPLVM provide a mapping from the latent space to the observation space, whose variables are assumed to be determined by the latent ones.

The formulation of GPVLM derives from Probabilistic Principal Component Analysis (PPCA), being a non-linear generalization of it, in particular from Dual PPCA models. Thus, in GPVLMs, the inner product kernel is substituted for a non-linear covariance function, and the marginal likelihood function $p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})$ can be expressed as:

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{d=1}^{D} p(\mathbf{y}_{:,d}|\mathbf{X}, \boldsymbol{\theta}), \quad (9)$$

where $\mathbf{y}_{:,d}$ is the $d-$th column of the data matrix $\mathbf{Y}$, corresponding to the $d-$th dimension, and $\mathbf{y}_{:,d}|\mathbf{X}, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{y}_{:,d}|\mathbf{0}, \mathbf{K} + \sigma^2_{noise} I)$. In order to train the GPLVM, a maximum a posteriori estimation of $\mathbf{X}$ must be performed, maximizing Eq. (9) with respect to the latent variable values, and to kernel and noise parameters $\boldsymbol{\theta}$. As a result, GPLVM allows for predicting higher-dimensional variables $\mathbf{y}$ from lower dimensional ones $\mathbf{x}$, by using Eq. (4).

D. Covariate Gaussian Process Latent Variable Models

Contextual GPLVM [18] is an extension of GPLVM specially thought for cases in which there exists meaningful varying contextual information, registered in a covariate matrix $\mathbf{S} \in \mathbb{R}^{N \times C}$, for each observation $\mathbf{y}$. In such cases, c-GPLVM is aimed at learning a covariate-adjusted representation of $\mathbf{y}$, and hence can be used to modulate the response of the regression model according to the covariate data, which can be continuous as well as discrete. Therefore, c-GPLVM mappings are defined on the joint space of $\mathbf{X}$ and $\mathbf{S}$.

Thus, the GPVLM formulation is adapted to fix the input components that concern the context variables. For this purpose, different joint kernels are available in literature, depending on the form of interaction assumed between latent and covariate inputs. In this work, we considered the additive and product kernels defined on the joint space, detailed in [18]:

$$k^{\text{add}}((\mathbf{x_i}, \mathbf{s_i}), (\mathbf{x_j}, \mathbf{s_j})) = k^x(\mathbf{x_i}, \mathbf{x_j}) + k^s(\mathbf{s_i}, \mathbf{s_j}) \quad (10)$$

$$k^{\text{pro}}((\mathbf{x_i}, \mathbf{s_i}), (\mathbf{x_j}, \mathbf{s_j})) = \frac{\sigma^2_{xs}}{\sigma^2_x \sigma^2_s} \cdot k^x(\mathbf{x_i}, \mathbf{x_j}) \cdot k^s(\mathbf{s_i}, \mathbf{s_j}) \quad (11)$$

where k^x and k^s are squared exponential ARD kernels, see Eq. (3), and σ^2_x, σ^2_s, σ^2_{xs} are defined accordingly.

This construction makes it possible to work with very low dimensional spaces, given the advantage of using known covariate information. This result is crucial, as it allows to search solutions in significantly lower spaces, speeding convergence, as well as to impose covariate conditions on proposed solutions. Furthermore, it makes also possible to handle data with partially missing or censored covariate information.

E. Bayesian Optimization and Upper Confidence Bound

BO approaches focus on finding the extrema of objective functions that are either expensive to evaluate, present no closed-form expression, or have unknown derivatives and convexity properties. Under these assumptions, these methods have proved to be among the most sample-efficient approaches [27], which is a main goal of our work, and have been successfully used for several robotics applications, e.g. in [28].

These techniques comprise two elements: a stochastic surrogate model fitting the target function, and an acquisition function defined in a search space $\boldsymbol{\Omega_X} \subset \mathbb{R}^{N \times Q}$. On the one hand, the surrogate model leverages the information of collected observations to derive a posterior distribution from a prior distribution by means, for example, of GPs, as in Sec. II-B. Thus, surrogate models provide also useful information about prediction uncertainty, the predictive variance, typically higher in unexplored areas.

On the other hand, the acquisition function uses the surrogate model results to assess the usefulness of evaluating each point of the search space, placing value on both unexplored and promising areas, as they are more likely to have higher objective function values, involving a trade-off between exploration and exploitation. Therefore, as a result of the maximization of the acquisition function, a point of the search space is proposed to be the next sample to evaluate through the objective function.

Upper Confidence Bound is a very straightforward [9] and practical acquisition function method [29], defined by:

$$\text{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \kappa\sigma(\mathbf{x}), \quad (12)$$

where κ is a parameter (left to the user) that sets the importance of exploration versus exploitation. The new samples x^{new} are then generated as:

$$\mathbf{x}^{\text{new}} = \text{argmax}_{\mathbf{x} \in \boldsymbol{\Omega}_X} \text{UCB}(\mathbf{x}) \quad (13)$$

This method results in choosing to sample the point that presents the highest mean plus κ standard deviation values on the surrogate function model.

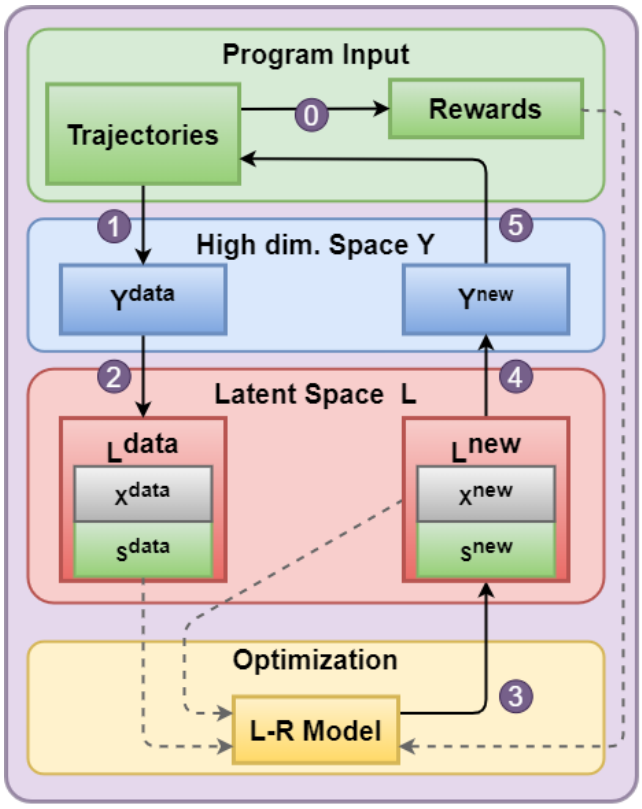

Fig. 2: Global scheme of the proposed approach. Trajectories are evaluated (0) and fitted (1) into data vectors $\mathbf{Y}^{\text{data}}$ as MP parameters. C-GPLVM (2) finds a latent data representation with variables $\mathbf{L}^{\text{data}}$, including free latent variables $\mathbf{X}^{\text{data}}$ and fixed context variables $\mathbf{S}^{\text{data}}$. These data, together with their corresponding reward values, are used to build a surrogate model of the reward function in the latent space, which can estimate the reward directly from the latent space. Moreover, UCB exploration is used (3) to generate new samples in the latent space, which are then used to predict (4) their respective high-dimensional space projection, and then executed (5) as trajectories and evaluated. The outputs of these evaluations and their generators in the latent space are sent back to the surrogate model of the reward, which will be updated.

III. Proposed Method

As mentioned in the introduction, we propose an approach that combines c-GPLVM [18], applied to the parameter space of a linear policy controller model such as an MP, with BO in the joint latent space, consisting of free latent and fixed context variables, in order to learn high-dimensional robot motion policies within very few samples.

In the first place, given a set of N trajectory demonstrations $\boldsymbol{\tau}_n = \{\mathbf{z}_j^n\}$ and covariate vectors $\boldsymbol{s}_n$, with indexes $n = 1..N$ and timesteps $j = 1..N_t$, we fit each trajectory to MP parameters $\boldsymbol{\omega}_n$ using least-squares according to Eq. (1). These weighting vectors are then put together as the rows of our data matrix $\mathbf{Y}$, so $D = N_f \cdot N_d$. As a PS algorithm, we aim at learning a surrogate of a black-box reward function from the evaluations of trajectories by means of BO:

$$\begin{aligned} R : \mathbf{Y} \subset \mathbb{R}^D &\longrightarrow \mathbb{R} \\ \mathbf{y} &\longmapsto R(\mathbf{y}) \end{aligned} \tag{14}$$

However, BO does not work properly with high-dimensional search spaces, and hence we make use of DR.

Algorithm 1

Input:
Trajectory data τ_{jl}^n , $n = 1..N$, $j = 1..N_t$, $l = 1..N_{\text{d}}$
Covariate data s_c^n , $n = 1..N$, $c = 1..C$
Demonstrated trajectories rewards R_n, $n = 1..N$
c-GPLVM free latent space dimension Q
MPs' kernel matrix $\boldsymbol{\Psi}$
Desired context data s_c^{des} , $c = 1..C$

1: Compute weights $\boldsymbol{\omega}_n$ with Eq.(1)
2: Assign $\mathbf{Y}_n \leftarrow \boldsymbol{\omega}_n$
3: Perform c-GPLVM($\mathbf{Y}_n, \mathbf{S}_n$), fixing $\mathbf{S}_n$, and obtain $\mathbf{X}_n$
4: Build $\mathbf{L}$-R Regression Model $f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$
5: Define search region $\boldsymbol{\Omega}_{\boldsymbol{L}} \subset \mathbb{R}^Q \times \{\boldsymbol{s}^{\text{des}}\}$
6: **for** $k = 1..N_{new}$ **do**
7: Define $\text{UCB}(\mathbf{x}, \mathbf{s}) = \mu_{k-1}(\mathbf{x}, \mathbf{s}) + \kappa\sigma_{k-1}(\mathbf{x}, \mathbf{s})$
8: Find new sample $(\mathbf{x}_k, \mathbf{s}^{\text{des}}) = \arg\max_{\mathbf{x}\in\boldsymbol{\Omega}_X} \text{UCB}(\mathbf{x}, \mathbf{s}^{\text{des}})$
9: Project $(\mathbf{x}_k, \mathbf{s}^{\text{des}})$ to $\tilde{\mathbf{y}}_k$ with Eq. (4)
10: Execute $\tilde{\mathbf{y}}_k$ and evaluate $R(\tilde{\mathbf{y}}_k(\mathbf{x}_k, \mathbf{s}^{\text{des}}))$
11: Update f, μ_k, and σ_k with $(\mathbf{x}_k, \mathbf{s}^{\text{des}})$ and $R(\mathbf{x}_k, \mathbf{s}^{\text{des}})$
12: **end for**

Contrary to other methods that perform DR directly in the space of degrees of freedom of the robot, as [20], [22], in our approach DR is applied in the parameter space of the MP. While the first approach is advantageous in that it provides qualitative information that is directly interpretable, our alternative enables to reduce further the dimensionality of the latent space, which is one of our main goals.

Therefore, a c-GPLVM is then fitted that maps the joint latent space of free latent variables and context variables $\mathbf{L} := \mathbf{X} \times \mathbf{S}$ of low dimension Q+C, to the high-dimensional observations space $\mathbf{Y}$, by fixing the context variables and optimizing the marginal log-likelihood function with respect to free latent variables, and to noise and kernel parameters. Thus, from Eq. (9) and Gaussian distribution properties, we can derive, (as in [24]):

$$\begin{aligned} &\log\, p(\mathbf{Y}|\mathbf{L}, \boldsymbol{\theta}) = \log \prod_{d=1}^{D} p(\mathbf{y}_{:,d}|\mathbf{L}, \boldsymbol{\theta}) \\ &= \sum_{d=1}^{D} -\frac{N}{2}\log\, 2\pi - \log\, |\mathbf{K}| - \frac{1}{2}\mathbf{y}_{:,d}^T\mathbf{K}^{-1}\mathbf{y}_{:,d}^T = \\ &= D(-\frac{N}{2}\log\, 2\pi - \log\, |\mathbf{K}|) - \sum_{d=1}^{D} \mathbf{y}_{:,d}^T\mathbf{K}^{-1}\mathbf{y}_{:,d} = \\ &= C - \sum_{d=1}^{D} \text{tr}\left(\mathbf{Y}^T\mathbf{K}^{-1}\mathbf{Y}^T\right) \end{aligned} \tag{15}$$

where C is a constant.

This optimization has been performed making use of tools provided by the GPy software framework, including the limited-memory BFGS optimizer. Once having performed this optimization, not only the set of parameters $\boldsymbol{\theta}$ are fitted, but also the c-GPLVM provides a proper latent space representation of data, from which we build a surrogate model for the reward function, by means of a Gaussian

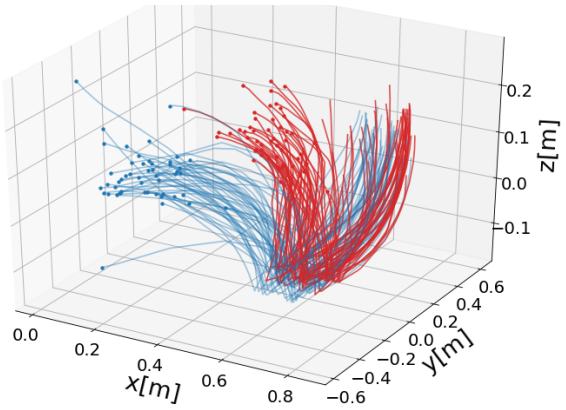

Fig. 3: Robot feeding trajectories obtained through kinesthetic teaching in the 3D space by two different instructors, represented in blue and red. Trajectories start at the points marked with a dot.

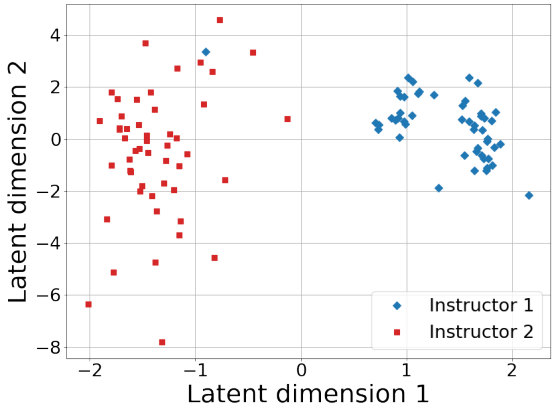

Fig. 4: Feeding data visualization in the latent space for first experiment. In blue, data corresponding to first instructor. In red, data corresponding to second instructor.

Process (see Sec. II-B):

$$\begin{aligned} \hat{R} : \mathbf{X} \times \mathbf{S} \subset \mathbb{R}^{Q+C} &\longrightarrow \mathbb{R} \\ (\mathbf{x}, \mathbf{s}) &\longmapsto \hat{R}(\mathbf{x}, \mathbf{s}) \end{aligned} \tag{16}$$

Subsequently, we make use of UCB (see Sec. II-E) to generate new sample candidates, according to the context requirement. For this purpose, the UCB uses the mean and variance provided by the GP surrogate model $\hat{R}$. As already mentioned in Sec. II-E, UCB suggests to evaluate points, according to the maximization of Eq. (12), in a certain search space $\mathbf{\Omega_L}$.

In this work, given a desired context vector $\mathbf{s}^{\text{des}}$, this search space has been defined as the Cartesian product of the minimum axis-aligned hyperrectangle that contains all free latent variables $\mathbf{\Omega_X} \subset \mathbb{R}^Q$, as suggested in [15], and the desired context vector, i.e., $\mathbf{\Omega_L} = \mathbf{\Omega_X} \times \{\mathbf{s}^{\text{des}}\}$. Thus, given some context requirements we fix the covariate vector $\mathbf{s}^{\text{des}}$, reducing the initial latent search space $\mathbf{\Omega_L}$ to a Q-dimensional subspace.

Furthermore, the exploration parameter κ in Eq. (12) has been fixed for simplification purposes ($\kappa = 1$) although less naive methods for selecting this parameter can be found in literature [16], [30].

Finally, the candidate selected by UCB $(\mathbf{x}, \mathbf{s}^{\text{des}})^{\text{new}}$ is reprojected to $\mathbf{Y}$ space, obtaining $\tilde{\mathbf{y}}(\mathbf{x}, \mathbf{s}^{\text{des}})^{\text{new}}$, and then decoded to a trajectory with Eq. (1), executed and evaluated, giving us the real value of the reward function $R(\mathbf{x}, \mathbf{s})$. This new sample, and the associated reward, will then be added to the surrogate model, that will be updated before generating new samples.

The process is repeated until convergence or a certain number of samples have been executed. From a computational point of view, it is noteworthy that the entire process runs in few seconds on a standard computer, therefore being negligible compared to real robot times of execution. Algorithm 1 displays the procedure of the proposed method, while Fig. 2 shows a more schematic view.

IV. Experimentation

We tested our method with two different experiments performed with a Barrett's WAM robot manipulator:

A. Feeding task

In this experiment, 100 feeding trajectories were demonstrated to the robot by two different instructors, each of them performing 50. Teachers guided the robot differently, from opposite positions, which involves one first categorical covariate variable $s_1 \in \{0, 1\}$ relative to style. In both cases, the robot goes from one initial to a final position, where a mannequin head is placed, getting food from one varying middle position on the table. The trajectories of its end-effector are showed in Fig. 3. Then, we complete the desired context vector with the coordinates of a specific middle point, by positioning the bowl at a particular place on the table, which is the objective point $\{s_2, s_3, s_4\} = \mathbf{o_p}$. Moreover, we define a reward function by calculating the Euclidean distance between the lowest-height point of each trajectory, the contact point $\mathbf{c_p}$, and $\mathbf{o_p}$:

$$R = -\text{dist}(\mathbf{c_p}, \mathbf{o_p})^2 \tag{17}$$

As input data, we used, asides from the covariate data matrix including style and contact points, the positions $\{x, y, z\}_{t=1..30}$ of the robot's end-effector to represent each trajectory, and 15 Gaussians per Cartesian dimension, resulting in a 45-dimensional parameter space. From this space, our c-GPLVM reduces dimension to 2+4 (free latent + context). A 2-D projection of these data can be visualized in Fig. 4, corresponding to free latent variables.

After that, we start the learning process by fitting the reward's surrogate model with a naive initial sample $(\mathbf{x}, \mathbf{s})_0 = (\mathbf{0}, \{s_1, \mathbf{o_p}\})$, and let the algorithm work to iteratively update the model by processing 50 new samples and evaluations through UCB. In Fig. 5, an example of learning curve for a random covariate vector is presented, compared with the results obtained with REPS. The plot shows that our method is impressively data-efficient, given that after only a dozen trials $-\log_{10}(-R) > 4$, which means a sub-centimeter precision for the contact point.

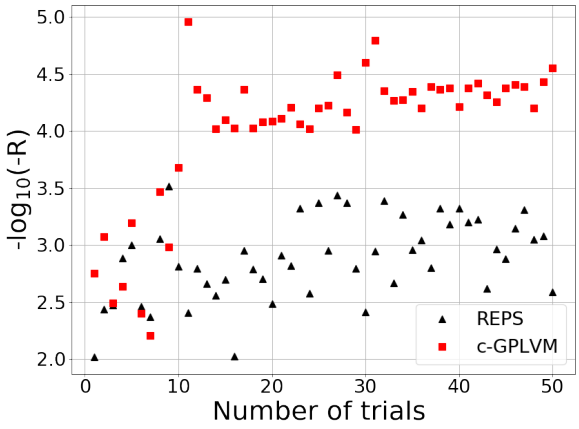

Fig. 5: A learning curve example of the first experiment, In red, the performance of our method, which is compared with REPS, in black.

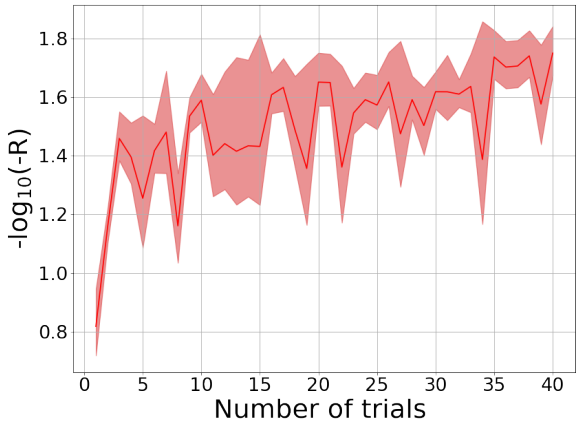

Fig. 6: Average learning curve for the second experiment, performed with five different leg poses. It shows the mean and the standard deviation of the performance. The fluctuations in both mean and standard deviation are totally justified by the fact that the algorithm is constantly exploring.

B. Shoe fitting

Shoe fitting (see Fig. 1) is the most complex task in which our algorithm that has been tested, as it implies learning in a 105-dimensional space, since we worked with 7 degrees of freedom and $N_f = 15$. Furthermore, here not only a final objective point position is important, but also the path matters, as the foot has to physically enter the shoe without colliding. For the same reason, the orientation of the end-effector, which holds the shoe, must also be considered. In this work, orientations have been encoded through quaternions of the form $\mathbf{q} = \{q_i, q_j, q_k, q_r\}$ as well as rotation matrices, changing between both equivalent systems when needed.

Therefore, we have made use of an HTC Vive Virtual Reality tracker system, which provides information on the leg pose, i.e., position and orientation, with respect to the robot base reference frame. This information is fixed during each robot trajectory, and hence it has been used as covariate vector $\mathbf{s} = \{x, y, z, q_i, q_j, q_k, q_r\}_{\text{leg}}$, being $C = 7$ in this case. Thus, 40 trajectories with different leg poses were demonstrated and the 105-dimensional MP parameter space could be reduced to 2 free latent plus 7 covariate dimensions, i.e., $Q = 2$ and $C = 7$.

Moreover, the tracker is not placed on the foot for obvious reasons, but rather on the other side of the leg, as shown in Fig. 1. Therefore, we needed to perform a reference change to know the desired pose of the end-effector, according to tracker signals. This was done with homogeneous transformation matrices [31]: Firstly, we manually fitted the shoe and measured the poses of the HTC tracker, which was previously calibrated, and the robot end-effector, by means of forward kinematics, to find the transformation between them after a successful shoe-fitting action. Such transform allows us to find the end-effector's desired pose given the leg pose, for any reading of the tracker attached to the leg. As a result, we can compute the desired final pose of the end-effector $\mathbf{p}_{\text{F}}^{\text{des}}$, leveraging the expression in transformation matrix form:

$$\mathbf{p}_{\text{F}}^{\text{des}}(\mathrm{M}_{\text{tracker|base}}^{\text{des}}) = \mathrm{M}_{\text{tracker|base}}^{\text{des}} \cdot [\mathrm{M}_{\text{tracker|base}}^{\text{fitted}}]^{-1} \cdot \mathrm{M}_{\text{e-e|base}}^{\text{fitted}} \tag{18}$$

where $\mathrm{M}_{\text{ref.1|ref.2}}$ is the transformation matrix from reference 1 to reference 2. In Eq. (18), $[\mathrm{M}_{\text{tracker|base}}^{\text{fitted}}]^{-1} \cdot \mathrm{M}_{\text{e-e|base}}^{\text{fitted}}$ is fixed and it was measured only once for a fitted shoe. Similarly, we computed a desired approaching pose of the end-effector $\mathbf{p}_{\text{A}}^{\text{des}}$, in order to make possible the entrance of the feet in the shoe. Such via-point $\mathbf{p}_{\text{A}}^{\text{des}}$ was imposed at a certain moment of the trajectory in order to make sure the foot is inserted into the shoe. Therefore, the reward function has been calculated with the Euclidean distances between desired and real, approaching and final poses:

$$\begin{aligned} R &= -\text{dist}(\mathbf{p}_{\text{F}}^{\text{des}}, \mathbf{p}_{\text{F}}^{\text{real}})^2 - \text{dist}(\mathbf{p}_{\text{A}}^{\text{des}}, \mathbf{p}_{\text{A}}^{\text{real}})^2 \\ &= -\sum_{i=1}^{7} (\mathrm{p}_{\text{F},i}^{\text{des}} - \mathrm{p}_{\text{F},i}^{\text{real}})^2 - \sum_{i=1}^{7} (\mathrm{p}_{\text{A},i}^{\text{des}} - \mathrm{p}_{\text{A},i}^{\text{real}})^2 \end{aligned} \tag{19}$$

In this experiment, we generated 40 new trajectories for 5 new random context goals. Logically, these new goals should not be too different to those demonstrated. The resulting average learning curve is presented in Fig. 6. For safety reasons, not all the new created trajectories were physically performed, as it might cause damage to the robot or the leg. However, once the algorithm had converged, we physically executed the resulting optimized trajectories with the WAM robot and proved that the shoe fitting is then successful, as shown in the attached video, which can be also found in `https://youtu.be/Og7116jb-04`.

In this video, it is also shown that the robot does even learn to fit the heel tab of the shoe, i.e. its posterior border, which, curiously, was not an easy task for some instructors using only one hand. Moreover, in Fig. 6, we can appreciate how fast the algorithm learns despite the high-dimensionality of the task, as it takes only about ten attempts to reach reward values such that $-\log_{10}(-R) > 1.5$, which are empirically proved to be successful.

V. Conclusion

In this paper, we presented an approach at learning context-adaptive robot motion in a efficient manner. By using c-GPLVM, a non-linear DR technique, we reduce the dimension of the MP parameter space by more than one order of magnitude, allowing the learning agent to sample in a low-dimensional manifold, and thus converging to a context-adaptable good solution with very few samples. Fig. 5 and 6 show that, after a given set of samples, the agent is capable of generalizing the model to any context after few real-robot samples. These samples, chosen by the UCB method, *fill the gaps* in the collected data in order to a better generalization.

As a future work, we intend to perform updates of c-GPLVM during the learning process, to consider task modulation according to time-varying context data, to create artificial data when needed by means of random variations of the context vector, to complete reward information with user ratings, and to use GPLVM extensions such as Bayesian GPLVM [32], [33]. Furthermore, this work, together with the one presented in [14], is planned to be extended and improved, including a more exhaustive evaluation of the algorithm, comparing it with several of the aforementioned state-of-the-art methods.

In conclusion, while other contextual policy search methods require hundreds or more real robot executions until converging to high quality policies, and other data-efficient state-of-the-art approaches are model-based or cannot deal with contextual case, in this paper we proposed a micro-data method specially devised for contextual case, without modelling the dynamics of the robot, which is proved to be useful for learning in some complex robotic tasks.

References

[1] B. D. Argall, S. Chernova, M. Veloso and B. Browning, “A survey of robot learning from demonstration". *The International Journal of Robotics Research*, vol. 32, pp. 1238-1274, 2013.

[2] J.Kober, J.A. Bagnell and J. Peters, "Reinforcement Learning in Robotics: A Survey". *Robotics and Autonomous Systems*, vol. 57, pp. 469-483, 2009.

[3] M. P. Deisenroth, G. Neumann and J. Peters, “A survey on Policy Search for Robotics". *Foundations and Trends in Robotics*, vol. 2, pp. 1-142, 2013.

[4] A. Kupcsik, M. P. Deisenroth, J. Peters, A. P. Loh, P. Vadakkepat and G. Neumann, “Model-based contextual policy search for data-efficient generalization of robot skills". *Artificial Intelligence*, vol. 247, pp. 415-439, 2015

[5] A. Fabisch and J. H. Metzen, "Active Contextual Policy Search". *Journal of Machine Learning Research*, vol. 15, no. 97, pp. 3371-3399, 2014.

[6] R. Pinslerk, P. Karkus A. Kupcsik D. Hsu and W. S. Lee, "Factored Contextual Policy Search with Bayesian optimization". *International Conference on Robotics and Automation (ICRA)*, pp. 7242-7248, 2019.

[7] P. Klink, H. Abdulsamad, B. Belousov and J. Peters, "Self-Paced Contextual Reinforcement Learning". *Conference on Robot Learning (CoRL)*, 2019.

[8] J. Peters, K. Mülling and Y. Altün, "Relative Entropy Policy Search". *National Conf. on Artificial Intelligence*, track 15, pp. 182-189, 2011.

[9] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J. B. Mouret, "A survey on policy search algorithms for learning robot controllers in a handful of trials". arXiv preprint arXiv:1807.02303, 2018.

[10] M. Deisenroth and C. E. Rasmussen. "PILCO: A model-based and data-efficient approach to policy search". *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465-472, 2011.

[11] K. Chatzilygeroudis, R. Rama, R. Kaushik, D. Goepp, V. Vassiliades and J. B. Mouret, "Black-box data-efficient policy search for robotics". *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 51-58, 2017.

[12] C. Daniel, G. Neumann, O. Kroemer and J. Peters, "Hierarchical relative entropy policy search". *Journal of Machine Learning Research*, vol 17, pp. 1-50, 2016.

[13] C. Daniel, M.Viering, J. Metz, O.Kroemer and J. Peters, “Active Reward Learning". *Proceedings of Robotics: Science and Systems (RSS)*, 2014.

[14] J. A. Delgado-Guerrero, A. Colomé and C. Torras, "Sample-efficient robot motion learning using Gaussian process latent variable models", *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[15] E. Brochu, V. M. Cora, N. de Freitas, "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning", arXiv preprint arXiv:1012.2599, 2010.

[16] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: no regret and experimental design" *International Conference on Machine Learning (ICML)*, pp 1015-1022, 2010.

[17] C. E. Rasmussen and C. K. I. Williams, "Gaussian Processes for Machine Learning", *the MIT Press*, 2006. ISBN 026218253X.

[18] K. Märtens, K. R. Campbell, C. Yau, "Decomposing feature-level variation with Covariate Gaussian Process Latent Variable Models", *Proceedings of the 36th International Conference on Machine Learning (PLMR-97)*, pp. 4372-4381, 2019.

[19] S. Bitzer and S. Vijayakumar, ”Latent spaces for dynamic movement primitives." *IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 574 - 581, 2009.

[20] A. Colomé and C. Torras. "Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes". *IEEE Transactions on Robotics*, vol. 34, no .3, pp. 602-615, 2018.

[21] A. Colomé, G. Neumann, J. Peters and C. Torras. "Dimensionality reduction for probabilistic movement primitives", *IEEE-RAS Humanoid Robots*, pp. 794-800, 2014.

[22] K. S. Luck, G. Neumann, E. Berger, J. Peters, and H. Ben Amor, ”Latent space policy search for robotics.” *IEEE/RSJ Int. Conf. on Intelligent Robots (IROS)*, pp. 1434-1440, 2014.

[23] A. Colomé and C. Torras. "Dimensionality reduction in learning Gaussian mixture models of movement primitives for contextualized action selection and adaptation", *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3922-3929, 2018.

[24] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models", *Journal of machine learning research*, vol. 6, no. Nov, pp. 1783–1816, 2005.

[25] C. Bishop. Pattern recognition and machine learning. Springer, 2007. ISBN 0387310738.

[26] N. Lawrence, "Gaussian Process Latent Variable Models for Visualisation of High Dimensional" *International Conference on Neural Information Processing Systems*, 2004.

[27] J. Mockus. "Application of Bayesian approach to numerical methods of global and stochastic optimization", *Journal of Global Optimization*, vol. 4, no. 4, pp. 347-365, 1994.

[28] D. J. Lizotte, T. Wang, M. H Bowling, and D. Schuurmans, "Automatic gait optimization with gaussian process regression". *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pp. 944-949, 2007.

[29] P. Hennig and C.J. Schuler. "Entropy search for information-efficient global optimization", *Journal of Machine Learning Research*, vol. 13, no. Jun, pp.1809-1837, 2012.

[30] S. Grünewälder, J.-Y. Audibert, M. Opper and J. Shawe-Taylor, "Regret Bounds for Gaussian Process Bandit Problems". *Journal of Machine Learning Research*, no 9. pp 273-280, 2010.

[31] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, "Robotics: Modelling, Planning and Control". Springer, 2009. ISBN 978-1-84628-641-4.

[32] M. K. Titsias and N. D. Lawrence, "Bayesian Gaussian Process Latent Variable Model". *International Conference on Artificial Intelligence and Statistics*, vol. 9 of JMLR:W&CP 9, 2010.

[33] P. Li, S. Chen, "A review on Gaussian Process Latent Variable Models". *CAAI Transactions on Intelligence Technology*, vol. 1, no. 4, pp. 366-376, 2016.