Learning from demonstration with model-based Gaussian process

Noémie Jaquier[1] **David Ginsbourger**[1,2] **Sylvain Calinon**[1]
[1]Idiap Research Institute
Rue Marconi 19, 1920 Martigny Switzerland
name.surname@idiap.ch
[2]IMSV, University of Bern
Alpeneggstrasse 22, 3012 Bern, Switzerland
david.ginsbourger@stat.unibe.ch

Abstract: In learning from demonstrations, it is often desirable to adapt the behavior of the robot as a function of the variability retrieved from human demonstrations and the (un)certainty encoded in different parts of the task. In this paper, we propose a novel multi-output Gaussian process (MOGP) based on Gaussian mixture regression (GMR). The proposed approach encapsulates the variability retrieved from the demonstrations in the covariance of the MOGP. Leveraging the generative nature of GP models, our approach can efficiently modulate trajectories towards new start-, via- or end-points defined by the task. Our framework allows the robot to precisely track via-points while being compliant in regions of high variability. We illustrate the proposed approach in simulated examples and validate it in a real-robot experiment.

Keywords: Imitation learning, Gaussian process, Gaussian mixture model

1 Introduction

In the context of learning from demonstrations (LfD), robot motions can be generated from demonstrated trajectories using various probabilistic methods, e.g. Gaussian mixture regression (GMR) [1], dynamical movement primitives (DMP) [2], probabilistic movement primitives (ProMP) [3] kernelized movement primitives (KMP) [4] or Gaussian process regression (GPR) [5]. The covariance matrices of the prediction distributions computed by GMR, ProMPs and KMP encode the variability of the predicted trajectory. This variability, reflecting the dispersion of the data collected during the demonstrations, carries important information for the execution of the task. For example, the phases of the task in which a high precision is required, e.g. picking an object in a specific location, are characterized with a low variability, and vice-versa. During the reproduction, the variability is typically used to define robot tracking precision gains and permits the combination of different controllers [6]. However, the approaches encoding variability do not take into account the availability of data in the different phases of the task. Inversely, the covariance matrices of the prediction distribution of GPs correspond to the prediction uncertainty, which reflects the presence or absence of training data in different phases of the task. This uncertainty measurement has been used, for example, to modulate the behavior of the robot far from the training data [6] or to actively make requests for new demonstrations in unexplored regions of the input space [7].

In LfD, it is often desirable to precisely refine parts of the demonstrated trajectories (e.g. due to changes in the environment), while maintaining the general trajectory shape (mean and variability) as in the demonstrations. It is also desirable to adapt the behavior of the robot, e.g. its compliance at different phases of the tasks, as a function of the variability of the demonstrations or the presence of (un)certainty in the reproduction. As none of the aforementioned methods provide both information features simultaneously, several approaches have been developed to take into account prediction uncertainty and variability. In [5], the reproduced trajectories are computed as a product of the predictions of local GPs, obtained by clustering the input space similarly to the approach of [8]. Therefore, by adapting the parameters of each GP, the resulting uncertainty is adapted as a function of the variability of the different phases of the demonstrations. In [9], the prediction uncertainty and variability are inferred separately. The trajectories are predicted using a combination of GP and dynamical movement primitives (DMP), therefore providing uncertainty measurement. On the other hand, the variability in the reproduction is determined by inferring the components of the corresponding covariance matrix with GPs.

3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan.

In this paper, we propose an approach that aims at encapsulating the variability information of the demonstrations in the prediction uncertainty. We take inspiration from multi-output Gaussian processes (MOGPs) under the linear model of coregionalization (LMC) assumption to design a non-stationary, multi-output kernel based on GMR. In contrast with the aforementioned approaches, both variability and uncertainty information are encoded in a single GP. Moreover, we define the prior mean of the process as equal to the mean provided by GMR. This permits to ignore the training data in the generation of new trajectories and to consider only via-points constraints as observed data, therefore alleviating the computational cost of the GP. Moreover, setting the tracking precision as a function of the retrieved covariance allows us to demand the robot to precisely track the via-points while lowering the required tracking precision in regions of high variability.

The remainder of the paper is organized as follows. GMR and GPR are reviewed and compared in Section 2. The proposed GMR-based GP is introduced in Section 3 and validated in a real-robot experiment in Section 4. Finally, Section 5 presents a discussion on similarities and differences of the proposed approach compared to other probabilistic methods, notably ProMP and KMP.

2 Background

2.1 Gaussian mixture regression

Gaussian mixture regression (GMR) exploits the Gaussian conditioning theorem to estimate the distribution of output data given input data [10, 11]. A Gaussian mixture model (GMM) is first estimated to encode the joint distribution of input and output datapoints, e.g., with an Expectation-Maximization (EM) algorithm. The output given observed input is then predicted via a linear combination of conditional expectations. Hence, GMR does not fit the regression function directly, but relies instead on the learned joint distribution.

We denote $\boldsymbol{X}$ and $\boldsymbol{Y}$ random vectors of input and corresponding output data, respectively, and by $\boldsymbol{x}$ and $\boldsymbol{y}$ arbitrary realizations of them. In a GMM with C components, the joint distribution of $(\boldsymbol{X}^\top, \boldsymbol{Y}^\top)^\top$ is encoded by

$$\begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix} \sim \sum_{\ell=1}^{C} \pi_\ell \, \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_\ell^{\mathcal{X}} \\ \boldsymbol{\mu}_\ell^{\mathcal{Y}} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_\ell^{\mathcal{X}} & \boldsymbol{\Sigma}_\ell^{\mathcal{X}\mathcal{Y}} \\ \boldsymbol{\Sigma}_\ell^{\mathcal{Y}\mathcal{X}} & \boldsymbol{\Sigma}_\ell^{\mathcal{Y}} \end{pmatrix} \right), \tag{1}$$

with π_ℓ, $\boldsymbol{\mu}_\ell$ and $\boldsymbol{\Sigma}_\ell$ the mixing coefficient (prior), mean and covariance matrix of the ℓ-th component.

GMR computes the conditional distribution of the GMM joint distribution to infer the output vector corresponding to a given input vector. The resulting multimodal distribution possesses second order moments that can be calculated from the conditional means and covariances of the multivariate Gaussian distributions associated with individual components using the laws of total mean and covariance, so that

$$\hat{\boldsymbol{y}}^{\mathrm{M}}(\boldsymbol{x}) = \sum_{\ell=1}^{C} h_\ell(\boldsymbol{x}) \hat{\boldsymbol{y}}_\ell(\boldsymbol{x}) \;\text{ and }\; \hat{\boldsymbol{\Sigma}}^{\mathrm{M}}(\boldsymbol{x}) = \sum_{\ell=1}^{C} h_\ell(\boldsymbol{x}) \tilde{\boldsymbol{\Sigma}}_\ell(\boldsymbol{x}) - \; \hat{\boldsymbol{y}}^{\mathrm{M}}(\boldsymbol{x}) (\hat{\boldsymbol{y}}^{\mathrm{M}}(\boldsymbol{x}))^\top, \tag{2}$$

with componentwise conditional means and covariances

$$\hat{\boldsymbol{y}}_\ell(\boldsymbol{x}) = \boldsymbol{\mu}_\ell^{\mathcal{Y}} + \boldsymbol{\Sigma}_\ell^{\mathcal{Y}\mathcal{X}} \boldsymbol{\Sigma}_\ell^{\mathcal{X}\,-1} \left(\boldsymbol{x} - \boldsymbol{\mu}_\ell^{\mathcal{X}}\right) \;\text{ and }\; \hat{\boldsymbol{\Sigma}}_\ell = \boldsymbol{\Sigma}_\ell^{\mathcal{Y}} - \boldsymbol{\Sigma}_\ell^{\mathcal{Y}\mathcal{X}} \boldsymbol{\Sigma}_\ell^{\mathcal{X}\,-1} \boldsymbol{\Sigma}_\ell^{\mathcal{X}\mathcal{Y}},$$

and $\tilde{\boldsymbol{\Sigma}}_\ell(\boldsymbol{x}) = \hat{\boldsymbol{\Sigma}}_\ell + \hat{\boldsymbol{y}}_\ell(\boldsymbol{x}) (\hat{\boldsymbol{y}}_\ell(\boldsymbol{x}))^\top$. The so-called responsability h_ℓ of component ℓ are computed in closed form as

$$h_\ell(\boldsymbol{x}) = \frac{\pi_\ell \, \phi\left(\boldsymbol{x}; \boldsymbol{\mu}_\ell^{\mathcal{X}}, \boldsymbol{\Sigma}_\ell^{\mathcal{X}}\right)}{\sum_{i=1}^{C} \pi_i \, \phi\left(\boldsymbol{x}; \boldsymbol{\mu}_i^{\mathcal{X}}, \boldsymbol{\Sigma}_i^{\mathcal{X}}\right)}, \tag{3}$$

where $\phi\left(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ stands for the probability density function at point $\boldsymbol{x}$ of the multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The computational complexity of GMR is mainly dependent on the number of GMM components as it governs the dimensionality of the maximum likelihood problem usually tackled by Expectation-Maximization. Moreover, the number of GMM components is the only parameter that needs to be specified and can be estimated online, e.g., with a Bayesian nonparametric approach [12]. Therefore, GMR is well adapted for real-time application and its simplicity allows it to be combined easily to other complementary approaches.

Figure 1a shows an example of application of GMR. The training dataset consists of 5 demonstrations of a two-dimensional time-driven trajectory. A GMM ($C = 6$) is first trained to encode the joint distribution of the inputs t and outputs $\boldsymbol{y}$. The conditional distribution of $\boldsymbol{y}$ given t is inferred by GMR. The covariance $\hat{\boldsymbol{\Sigma}}^{\mathrm{M}}(\boldsymbol{x})$ of the distribution encodes the variability of the demonstrations. The output distribution is extrapolated in the absence of training data ($t > 2$).

2.2 Gaussian process regression

Gaussian processes (GPs) form a class of probabilistic models that aims at learning a deterministic input-output relationship, up to observation noise, based on a Gaussian prior over potential objective functions. In the noiseless GP framework, the output y is hence typically seen as a function of a controlled input $\boldsymbol{x}$. Randomness comes in an instrumental way as the function $y(\boldsymbol{x})$ is assumed to be one realization of a Gaussian random process or random function denoted $Y(\boldsymbol{x})$. Predictions of the objective function are then made by relying on the conditional distribution of $Y(\boldsymbol{x})$ knowing that $Y(\boldsymbol{x}_n^{(\mathrm{o})})$ coincides with the observed outputs $\boldsymbol{y}_n^{(\mathrm{o})}$ at their corresponding observation inputs $\boldsymbol{x}_n^{(\mathrm{o})}$.

Multi-output Gaussian processes (MOGPs) generalize GPs to vector-valued output by predicting jointly the output components (see [13] for a review). Therefore, MOGPs exploits the potential relation between the output components, which are not taken into account if predictions are computed separately for each dimension. Similarly to standard GP, the vector-valued objective function is modeled in MOGP with a vector-valued GP ($\boldsymbol{Y}(\boldsymbol{x})$), inducing finite-dimensional prior distributions $\boldsymbol{Y}(\boldsymbol{x}_{1:N}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}_{1:N}}, \boldsymbol{K}_{\boldsymbol{x}_{1:N}})$ for any arbitrary set of inputs $\boldsymbol{x}_{1:N}$. We here denote $\boldsymbol{\mu}_{\boldsymbol{x}_{1:N}} = \boldsymbol{\mu}(\boldsymbol{x}_{1:N})$ and $\boldsymbol{K}_{\boldsymbol{x}_{1:N}} = \boldsymbol{k}(\boldsymbol{x}_{1:N}, \boldsymbol{x}_{1:N})$ the corresponding mean vector and covariance matrix, where $\boldsymbol{\mu}$ and $\boldsymbol{k}$ stand for the mean function and cross-covariance kernel of the MOGP and $\boldsymbol{\mu}_{\boldsymbol{x}} \in \mathbb{R}^D$, $\boldsymbol{K}_{\boldsymbol{x}} \in \mathbb{R}^{D\times D}$ with D the output dimension.

Denoting $\boldsymbol{y}_{1:N}^{(\mathrm{o})}$ the observed realization of $\boldsymbol{Y}(\boldsymbol{x}_{1:N}^{(\mathrm{o})}) + \boldsymbol{\varepsilon}$, the posterior distribution follows by Gaussian conditioning

$$\boldsymbol{Y}(\boldsymbol{x}) | \boldsymbol{y}_{1:N}^{(\mathrm{o})} \sim \mathcal{N}(\hat{\boldsymbol{y}}^{\mathrm{P}}(\boldsymbol{x}), \hat{\boldsymbol{\Sigma}}^{\mathrm{P}}), \tag{4}$$

with conditional mean and covariance functions given by

$$\hat{\boldsymbol{y}}^{\mathrm{P}}(\boldsymbol{x}) = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}_{1:N}^{(\mathrm{o})}} \left(\boldsymbol{K}_{\boldsymbol{x}_{1:N}^{(\mathrm{o})}} + \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}} \right)^{-1} (\boldsymbol{y}_{1:N}^{(\mathrm{o})} - \boldsymbol{\mu}_{\boldsymbol{x}_{1:N}^{(\mathrm{o})}}), \tag{5}$$

$$\hat{\boldsymbol{\Sigma}}^{\mathrm{P}}(\boldsymbol{x}) = \boldsymbol{K}_{\boldsymbol{x}} - \boldsymbol{K}_{\boldsymbol{x},\boldsymbol{x}_{1:N}^{(\mathrm{o})}} \left(\boldsymbol{K}_{\boldsymbol{x}_{1:N}^{(\mathrm{o})}} + \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}} \right)^{-1} \boldsymbol{K}_{\boldsymbol{x}_{1:N}^{(\mathrm{o})},\boldsymbol{x}}, \tag{6}$$

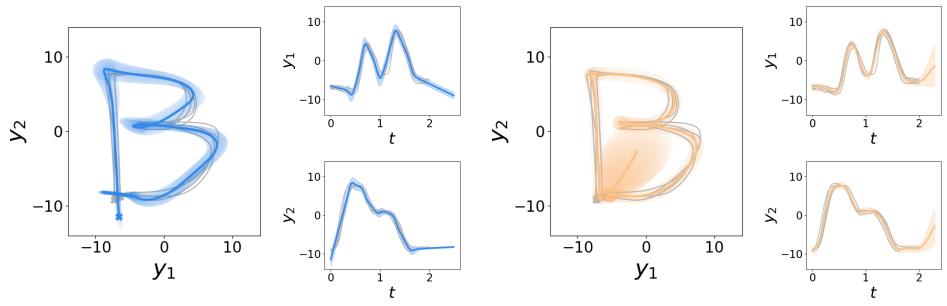
where $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$ is the covariance matrix of the observation noise $\boldsymbol{\varepsilon}$, which is assumed centered Gaussian and independent of the process $\boldsymbol{Y}$. The covariance $\hat{\boldsymbol{\Sigma}}^{\mathrm{P}}(\boldsymbol{x})$ expresses the prediction uncertainty for all components and between them. In typical cases, the further away the input data lies from the training dataset the larger the prediction variance, as illustrated in Fig. 1-(*right*). Moreover, the mean $\hat{\boldsymbol{y}}^{\mathrm{P}}(\boldsymbol{x})$ then converges to the prior mean $\boldsymbol{\mu}_{\boldsymbol{x}}$, equal to $\mathbf{0}$ in this case.

The class of covariance kernels that we consider in this paper is formulated as a sum of separable kernel functions generated under the linear model of coregionalization (LMC) assumption [14]. This class of kernel functions is often called separable as the dependencies between inputs and outputs are decoupled. Therefore, the kernel $\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}')$ between two input vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ is expressed as

$$\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{q=1}^{Q} \boldsymbol{\Upsilon}_q k_q(\boldsymbol{x}, \boldsymbol{x}'), \tag{7}$$

where the so-called coregionalization matrices $\boldsymbol{\Upsilon}_q \in \mathbb{R}^{D\times D}$ are positive semi-definite matrices representing the interaction among the output components. The choice of the scalar-valued kernels k_q and the design of the coregionalization matrices $\boldsymbol{\Upsilon}_q$ are crucial for the GP as they represent our prior knowledge about the function that is being learned.

Figure 1b shows an example of MOGP with the separable kernel (7) where $Q = 1$ and $k_q(\boldsymbol{x}, \boldsymbol{x}')$ is a Matérn kernel ($\nu = 5/2$), so that $k_q(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\right) \exp\left(-\frac{\sqrt{5}r}{\sigma_l}\right)$, where $r = \sqrt{(\boldsymbol{x} - \boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')}$ is the Euclidean distance between inputs and σ_f, σ_l are the variance and lengthscale parameters, respectively.



(a) (b)

Figure 1: Comparison between (*a*) GMR and (*b*) GPR. The training data are shown in light gray for all graphs. The mean is represented by a continuous line and the variance by a light tube around the estimate. The *left* graphs show the output trajectories estimated by GMR and GPR, respectively. The beginning of the trajectories is marked by a cross. The *right* graphs show the estimated trajectory for each output component as a function of the input t. Time and positions are given in seconds and centimeters, respectively.

3 GMR-based Gaussian Processes

In this section, we propose to combine GMR and GPR to form a GMR-based GP. The proposed approach takes advantage of the ability of GPs to encode various prior beliefs through the mean and kernel functions and allows the variability information retrieved by GMR to be encapsulated in the uncertainty estimated by the GP. Moreover, the proposed approach enjoys the properties of generative models, therefore new trajectories can be easily generated through sampling and conditioning.

3.1 GMR-based GPs formulation

We define the GMR-based GP as a GP with prior mean

$$\boldsymbol{\mu}(\boldsymbol{x}) = \hat{\boldsymbol{y}}^{\mathrm{M}}(\boldsymbol{x}), \tag{8}$$

and a kernel in the form of a sum of C separable kernels associated with the C components of the considered GMM

$$\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\ell=1}^{C} h_\ell(\boldsymbol{x}) h_\ell(\boldsymbol{x}') \hat{\boldsymbol{\Sigma}}_\ell k_\ell(\boldsymbol{x}, \boldsymbol{x}'). \tag{9}$$

The prior mean of Eq. (2) allows the GP to follow the GMR predictions far from training data. Moreover, the constructed GP is also covariance non-stationary due to its spatially-varying coregionalization matrices [15]. The input-dependent coregionalization matrices $h_\ell(\boldsymbol{x}_m) h_\ell(\boldsymbol{x}_n) \hat{\boldsymbol{\Sigma}}_\ell$ corresponding to this conception are determined by GMR (via (2), (3)). Alternatively, one can say that the GMR responsabilities h_ℓ weight the importance of the kernels $k_\ell(\boldsymbol{x}_m, \boldsymbol{x}_n)$ according to the proximity of the input data to the center of GMM components. Thus, the kernels associated to the centers close to the given input data are more relevant than distant centers. The covariance matrices $\hat{\boldsymbol{\Sigma}}_\ell$ allows the dependencies between the output data to be described for each GMM component. Note that both the coregionalization matrices and the number of separable kernels are determined by GMR. Therefore, the only parameters to determine are the hyperparameters of the kernels k_ℓ which can be estimated, for example, by maximizing the likelihood of the GP. Moreover, the variance parameters σ_f of the kernels k_ℓ are fixed to 1 as they are already scaled by the covariance matrices $\hat{\boldsymbol{\Sigma}}_\ell$. Thus, the estimation of hyperparameters is simplified compared to standard LMC.

Figure 2a shows the prior mean and 10 sample trajectories generated from the proposed GMR-based GP where k_ℓ are Matérn kernels ($\nu = 5/2$). The hyperparameters, namely the lengthscales $\sigma_{l,\ell}$ of the k_ℓ and the covariance of the noise $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \sigma_\epsilon \boldsymbol{I}$, were optimized by maximum likelihood estimation within the GPy framework [16]. Note that the prior mean of the process corresponds to the mean obtained by GMR in Figure 1a. Moreover, the prior uncertainty provided by the GMR-based GP is lower in the regions where the variability of the demonstrations is low, e.g. at the bottom of the straight vertical line of the *B* letter, and higher in the regions of higher variability, e.g. in the curves in the right part of the *B*.

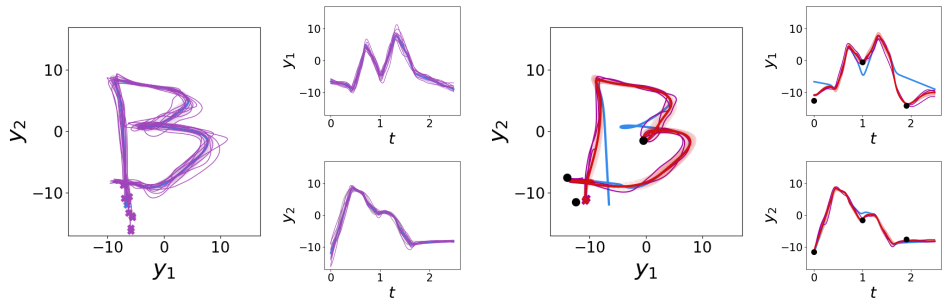


Figure 2: (*a*) Sample trajectories generated from GMR-based GP. The prior mean of the process and the sample trajectories are represented by continuous blue and purple lines, respectively. The covariance $\boldsymbol{K}_{t_{1:N}^{(o)}}$ of the process is represented by a light purple tube around the prior mean. (*b*) Sample and predicted trajectories generated from the posterior distribution of the GMR-based GP. The prior, sampled and predicted trajectories are represented by continuous blue, dark pink and red lines, respectively. The uncertainty of the prediction is represented by a light red tube around the predicted mean. Via-points, considered as observations for the GMR-based GP, are represented by black dots. The trajectories are extrapolated from training data for $t > 2$.

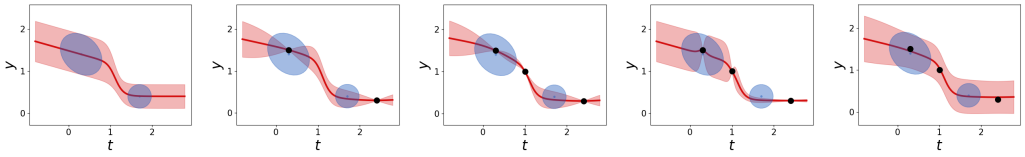
3.2 GMR-based GPs properties

A particularity of the presented GMR-based GP is that the information on the demonstrations distribution is included in the prior mean $\boldsymbol{\mu}_{\boldsymbol{x}}$ and covariance $\boldsymbol{K}_{\boldsymbol{x}}$ after determining the hyperparameters. Therefore, the training data can be ignored and our model can be conditioned uniquely on new observations. Figure 3a shows the mean and uncertainty recovered by a 1D-output GMR-based GP without any observation. The process was constructed based on a GMM with two components with k_ℓ defined as Matérn kernels ($\nu = 5/2$). The lengthscale parameters σ_l of the k_ℓ and the covariance of the noise of the process σ_ϵ are fixed as equal to 1 and $1e^{-4}$, respectively. Note that the distribution corresponds to the prior of the GMR-based GP, therefore the mean is exactly equal to the mean computed by GMR. Moreover, if a component ℓ is entirely responsible for a test datapoint so that $h_\ell = 1$, the corresponding uncertainty is equal to the conditional covariance of the component $\hat{\boldsymbol{\Sigma}}_\ell$ augmented with Σ_ϵ, as observed for $t < 0.6$ and $t > 1.8$ in Figure 3a. In the case where several components are responsible for the datapoint, its uncertainty is a weighted sum of the conditional covariances, as observed for the zone in between the two GMM components. Therefore, the prior uncertainty obtained by GMR-based GP without observation reflects the variability provided by GMR. However, note that the prior uncertainty of GMR-based GP is not equal to $\hat{\boldsymbol{\Sigma}}^{\mathrm{M}}$.

In the cases where it is desirable to adapt trajectories towards new start-, via- or end-points $(\boldsymbol{\xi}_v, \boldsymbol{\zeta}_v)$, those particular points are used to define a new set of observation inputs and outputs $(\boldsymbol{x}_{1:V}^{(o)}, \boldsymbol{y}_{1:V}^{(o)}) = (\boldsymbol{\xi}_{1:V}, \boldsymbol{\zeta}_{1:V})$ which is then used to infer the posterior distribution of the GMR-based GP with (4). Figures 3b and 3c show examples where 2 and 3, via-points were added to the trajectory. We observe that the mean of the process goes through the via-points and the uncertainty becomes very small in these locations. Note that conditioning a trajectory towards via-points with GMR alone is not straightforward due to the fact that covariance terms between two datapoints are equal to zero.

As in a standard GP, the predicted mean and uncertainty depend strongly on the kernel parameters. Moreover, one of the advantages of the GMR-based GP is that each kernel k_ℓ can be chosen individually and their parameters are determined separately. Therefore, different behaviors can be obtained as a function of the location in the input space, as shown by Figure 3d where the lengthscale parameters of the kernels corresponding to the left and right GMM component are equal to 0.1 and 5, respectively. Similarly to a standard GP, the noise of the process determines the behavior of the GMR-based GP at the via-points location. As shown by Figure 3e where $\sigma_\epsilon = 0.1$, the constraint of passing exactly through the via-points is alleviated and the mean of the GMR-based GP passes close-by the via-points while the uncertainty is equal to the noise of the process. Note that the noise parameter can also be defined separately for each kernel k_ℓ.

Figure 2b shows the predicted mean and corresponding uncertainty as well as three trajectories sampled from the posterior distribution of the GMR-based GP on the *B* trajectory. As explained



(a) (b) (c) (d) (e)

Figure 3: Example of time-driven 1D-trajectories predicted by GMR-based GP. The process was constructed based on a 2-components GMM, represented by blue ellipses. The estimate of GMR-based GPR is shown by a red continuous line with the corresponding uncertainty represented by a light tube around the mean. The initial observations are discarded after determining the hyperparameters. The lenghtscale parameters and noise covariance are fixed as $\sigma_l = 1$ and $\sigma_\epsilon = 1e^{-4}$, respectively. (*a*) The posterior distribution without any new observation is represented. (*b-c*) Two, respectively 3, via-points, represented with black dots, are added as observations of the GMR-based GP. (*d*) The lengthscale parameters are fixed as $\sigma_l = 0.1$, $\sigma_l = 5$ for the left and right GMM component, respectively. (*e*) The noise covariance of the process is fixed as $\sigma_\epsilon = 0.1$ ($\sigma_l = 1$).

previously, the initial demonstrations data were used for hyperparameters estimation but not incorporated as conditioning data and three via-points have been added as new observations. We observe that the estimate and the posterior trajectories are adapted to pass close to the via-points. In the zones far from the via-points, the predicted trajectory follows the prior mean of the process.

4 Experiments

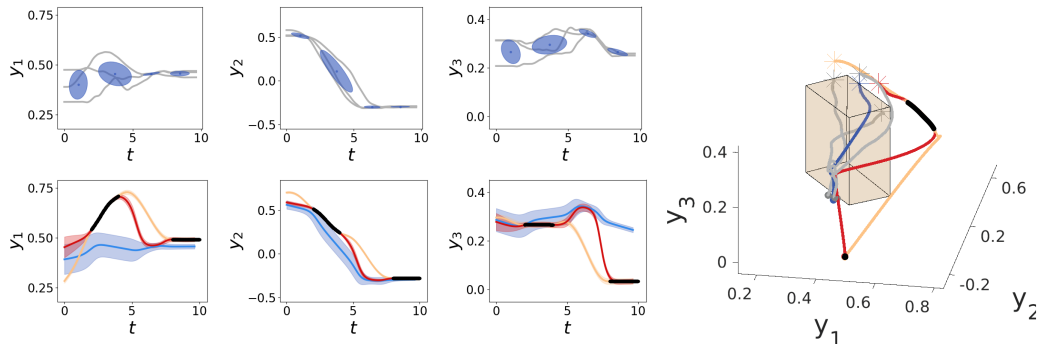
In this section, we evaluate the proposed approach in a peg-in-hole task achieved by the 7-DoF Franka Emika Panda robot. In the first part of the experiment, 3 demonstrations of the task were collected by kinesthetically guiding the robot to first approach a hollow cylinder and then insert the peg in it. For all the demonstrations, the hollow cylinder was placed 20 cm above the table. The collected data, encoding time t and Cartesian position $(y_1 \ y_2 \ y_3)^\top$, were time-aligned. We trained a GMM and determined the hyperparameters of a GMR-based GP, as well as a MOGP with the separable kernel (7) ($Q = C$) based on the time-driven demonstrated trajectories. Matérn kernels ($\nu = 5/2$) were chosen as individual kernels k_ℓ and k_q for the GMR-based GP and MOGP, respectively. The number of components of the GMM ($C = 4$) was selected by the experimenter. Figure 4a shows the demonstrated trajectories and corresponding GMM states.

In the second part of the experiment, an obstacle was added in between the initial position of the robot and the hollow cylinder. Moreover, the hollow cylinder was positioned directly on the table, i.e. 20 cm below its location during the demonstrations. Via-points were determined by the experimenter to modulate trajectories so that the robot avoids the obstacle in the desired manner and its final position corresponds to the new location of the hollow cylinder. The performances of the proposed GMR-based GP, the MOGP and GMR to reproduce the task in the modified environment were compared.

As explained in the previous section, the via-points were used to define a new set of observations for the GMR-based GP, while the original training data are discarded after inferring the hyperparameters. In the case of the MOGP, the mean and uncertainty of the reproduction considering V via-points are updated for each testing input $\boldsymbol{x}$ by conditioning the distribution (4) on the desired via-points. The mean and variability of the reproduction obtained by GMR can be updated in a similar way. However, as the covariances between different datapoints are not encoded in GMR, the generated trajectory is discontinuous. Therefore, we did not reproduce it with the robot and we show instead in the following graphs the GMR reproduction where no via-points are considered, whose mean corresponds to the prior mean of the GMR-based GP.

The task was reproduced using a linear quadratic regulator (LQR) controller tracking the trajectory predicted by GMR-based GPR, MOGP or GMR [1]. The required tracking precision was set as proportional to the inverse of the posterior covariance $\hat{\boldsymbol{\Sigma}}$ of the different methods. This information is exploited to demand a high precision tracking in the regions of the trajectories where high certainty (GMR-based GP and MOGP) or low variability (GMR) are observed, and vice-versa.

Figure 5 shows snapshots of the robot reproducing the peg-in-hole task using the proposed GMR-based GP (*top row*) and the MOGP (*bottom row*). We observe that the robot is able to circumvent



(a) (b)

Figure 4: (*a*)-*top* Demonstrated trajectories (in light gray) and corresponding GMM ($K = 4$) represented as blue ellipses. The Cartesian positions y_1, y_2, y_3, considered as outputs, are represented as a function of the time, considered as input. (*a*)-*bottom* Reproduced trajectories. The means of the trajectories generated by GMR-based GP, MOGP and GMR are represented by red, yellow and blue lines, respectively, with their respective covariance depicted by light tubes around the estimates. The via-points defined to modulate the trajectories generated by GMR-based GP and MOGP are depicted with black dots. (*b*) 3D representation of the reproduced trajectories by the robot. The beginnings of the demonstrated and reproduced trajectories are depicted by stars.

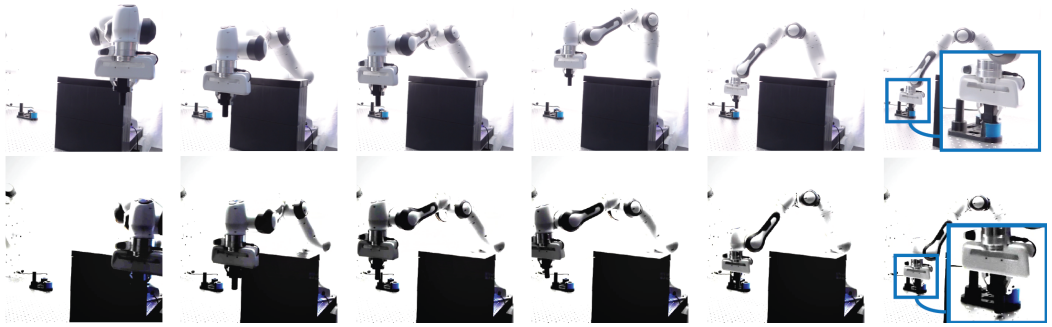


Figure 5: Snapshots of the robot reproducing the peg-in-hole task using GMR-based GP (*top row*) and MOGP (*bottom row*). Both methods allows the robot to circumvent the obstacle. However, the peg is successfully inserted in the hole with the GMR-based GP, while the robot fails to insert the peg with MOGP (observe that the blue hollow cylinder is behind the peg at the end of the trajectory).

the obstacle with both methods. However, the peg insertion fails when the MOGP is used, as the peg is located in front of the cylinder at the end of the trajectory. This is due to the fact that the trajectory generated by the MOGP straightly links the two zones characterized with via-points, while the GMR-based GP trajectory tends to follow the prior mean defined by GMR in between the two zones, as shown in Figure 4a-*bottom*. This behavior allows the robot to position the peg above the hole before approaching the cylinder and perform the insertion as demonstrated in the first phase of the experiment. Inversely, by using the MOGP, the robot approaches the hollow cylinder from the side, and therefore is unable to insert the peg. These different behaviors are illustrated in Figure 4b, where the 3D trajectories reproduced with the different methods are represented. In order for the MOGP to successfully reproduce the insertion, a supplementary via-point could be added prior to the insertion. However, this supplementary via-point is not needed by the GMR-based GP thanks to its prior mean.

Moreover, as shown in Figure 4a-*bottom*, the uncertainty computed by the MOGP is low along the whole trajectory, resulting in a rigid behavior of the robot for the whole reproduction. In contrast, the GMR-based GP ensures a high tracking precision in the two parts of the trajectory characterized by the via-points, while the robot can be more compliant elsewhere depending on the variability encoded by the GMR, notably at the beginning of the reproduced trajectory. A video of the experiment and source codes are available at https://sites.google.com/view/gmr-based-gp.

5 Discussion

By defining a prior mean as GMR and by encapsulating the variability of the demonstration in its uncertainty, the proposed GMR-based GP allows efficient reproductions of tasks learned by demonstration while adapting the learned trajectories towards new start-, via- or end-points. We discuss here similarities and differences between the proposed approach and other algorithms widely used to learn trajectories. Figures accompanying the discussion are displayed in Appendix A.

As briefly mentioned in the previous section, adapting trajectories with GMR is difficult as conditioning on via-points results in discontinuous trajectories and re-optimizing the underlying GMM to fulfill via-points constraints is not straightforward. In contrast, the trajectories can be easily adapted to go through start-, via- or end-points by conditioning on the desired observations in the case of GP and ProMP. Trajectories can also be adapted using DMP. However, DMP does not handle variation and uncertainty information. Moreover, as DMP and ProMP encode trajectories by relying on basis functions equally spaced in time, selecting appropriate basis functions becomes difficult with high-dimensional inputs. In contrast, kernel methods and GMR, for which GMM learns the organization of basis functions, generalize well to high-dimensional inputs. By using GP, the trajectories are modeled without considering the correlations between the output components. This problem can be alleviated by replacing GP by MOGP. Notice that the computational complexity of testing for the proposed GMR-based GP is importantly reduced compared to MOGP, as the set of observations used in the testing part is only composed of desired via-points, therefore resulting in a computational complexity of $\mathcal{O}(V^2D^2)$ instead of $\mathcal{O}(N^2D^2)$, with D the output dimension and $V \ll N$.

Overall, KMP shares strong similarities with the proposed GMR-based GP. Both approaches are kernel-based and can therefore cope with high-dimensional inputs. Moreover, both make use of GMR, to retrieve a reference trajectory in the case of KMP and to define the prior mean as well as kernel parameters for GMR-based GP. Therefore, the correlations between the output components are taken into account in both models and they predict full covariances for inferred outputs. Note that both approaches can make use of other algorithms than GMR to capture the distribution of the demonstrations.

Compared to KMP which can be related to kernel regression, the framework of GMR-based GP allows the representation of more complex behaviors, notably by defining priors for the process. Our approach benefits of the properties of generative models, allowing sampling of new trajectories from prior and posterior models (as shown in Fig. 2b), and is highly flexible due to the kernels k_ℓ that can be chosen individually, resulting in different behaviors of the model in the different regions of the input space. Moreover, GMR-based GP provide an uncertainty information encapsulating the variability in the variance parameter of the kernel, while KMP introduces the measure of the variability of the demonstrations as the covariance matrix of the observation noise. As a consequence, for the example of a robot tracking via-points, KMP will adapt the distribution according to the covariance, representing the variability of the demonstrations, given initially by GMR. In contrast, our approach allows us to set via-points that the robot is required to track precisely, where the covariance tends to zero due to GP properties. This is relevant for the case in which controller gains are set as a function of the observed covariance, as we can ensure high precision due to close-to-zero prediction variances.

6 Conclusion

This paper presented a new class of multi-output GP with non-stationary prior mean and kernel based on GMR. Our approach inherits of the properties of generative models and benefits of the expressiveness and versatility of GPs. Within this framework, the variability of the demonstrations is encapsulated in the prediction uncertainty of the designed GP. Correlations between the different output components are taken into account by the model. Moreover, the method takes advantage of the prior obtained from the demonstrations for trajectory modulation, considering only via-points constraints as observed data to generate new trajectories. Our framework allows a precise tracking of via-points while the compliance of the robot can be adapted as a function of the variability of the demonstrations in other parts of the trajectories. Extensions of this work will investigate more in details the properties and limits of the proposed approach. Moreover, we plan thorough comparisons between GMR-based GP and other approaches of interest, such as KMP. Finally, the proposed approach may also be considered in an active learning framework, where new datapoints are queried in regions of high uncertainty.

Acknowledgments

This work was supported by the Swiss National Science Foundation (SNSF/DFG project TACT-HAND).

References

[1] S. Calinon. A tutorial on task-parametrized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, 2016.

[2] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 763–768, 2009.

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

[4] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell. Kernelized movement primitives. *Intl. Journal of Robotics Research*, 38(7):833–852, 2019.

[5] M. Schneider and W. Ertel. Robot learning by demonstration with local Gaussian process regression. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 255–260, 2010.

[6] J. Silvério, Y. Huang, L. Rozo, S. Calinon, and D. G. Caldwell. Probabilistic learning of torque controllers from kinematic and force constraints. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 6552–6559, 2018.

[7] G. Maeda, M. Ewerton, T. Osa, B. Busch, and J. Peters. Active incremental learning of robot movement primitives. In *In Proc. of the 1st Annual Conf. on Robot Learning (CoRL)*, pages 37–46, 2017.

[8] D. Nguyen-Tuong, J. Peters, and M. Seeger. Local Gaussian process regression for real time online model learning and control. In *Advances in neural information processing systems 21*, pages 1193–1200, 2009.

[9] J. Umlauft, Y. Fanger, and S. Hirche. Bayesian uncertainty modeling for programming by demonstration. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 6428–6434, 2017.

[10] Z. Ghahramani and M. Jordan. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems (NIPS)*, volume 6, pages 120–127, 1994.

[11] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Artificial Intelligence Research*, 4:129–145, 1996.

[12] A. K. Tanwani and S. Calinon. Small-variance asymptotics for non-parametric online robot learning. *The International Journal of Robotics Research*, 38(1):3–22, 2019.

[13] M. A. Alvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

[14] P. Goovaerts. *Geostatistics for natural resources evaluation*. Oxford University Press, USA, 1997.

[15] A. E. Gelfand, A. M. Schmidt, S. Banerjee, and C. F. Sirmans. Nonstationary multivariate process modeling through spatially varying coregionalization. *TEST*, 13(2):263–312, 2004.

[16] GPy. GPy: A Gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

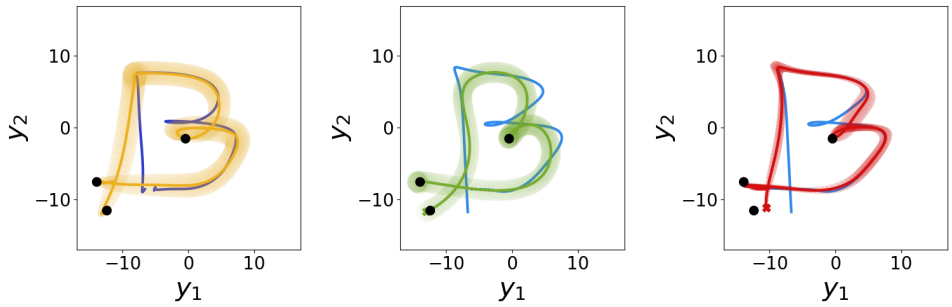
Appendices

A Comparison of GMR-based GP with ProMP and KMP

Figure 6 shows an example of modulated trajectories recovered by ProMP, KMP and GMR-based GP. The training data, consisting of 5 demonstrations of a two-dimensional time-driven trajectory, are identical to the training data of Figures 1 and 2. As in Figure 2b, via-points are represented by black dots. ProMP is evaluated with 20 Gaussian basis functions, while both KMP and GMR-based GP are based on a 6-components GMM.

As expected the three methods are able to generate trajectories passing through the via-points. As discussed in Section 5, ProMP encodes trajectories by relying on basis function equally spaced in time. In contrast, both KMP and GMR-based GP rely on GMR, for which GMM learns the organization of the basis functions. Note that the reference trajectory of KMP and the prior mean of GMR-based GP, depicted by a light blue line in Figures 6b and 6c are the same, as they both correspond to the mean of GMR.

We observe that the global shape of the trajectory is modified by introducing via-points with KMP. In contrast, GMR-based GP tends to follow the prior trajectory in the absence of via-points. However, this change of shape allows KMP to track more precisely the first via-point compared to ProMP and GMR-based GP. Moreover, in the case of KMP, the prediction variance at the via-points depends on the variability of the GMR, while it can be set close to zero with GMR-based GP. This is due to the fact that, in the case of KMP, the variability of the demonstrations, encoded by GMR, is introduced as the covariance matrix of the observation noise. In contrast, the tracking precision can be set independently with GMR-based GP, as shown by Figures 3c, 3e. The aforementioned difference between KMP and GMR-based GP can also be observed by comparing Figures 3 and 7. Moreover, as opposite to KMP, the behavior of GMR-based GP can be modulated in different regions of the input space due to the kernel k_ℓ that can be chosen individually (see Figure 3d).



(a) ProMP (b) KMP (c) GMR-based GP

Figure 6: Comparison of the predicted trajectories generated by (*a*) ProMP, (*b*) KMP and (*c*) GMR-based GP with three via-points. The mean is represented by a continuous line and the variance by a light tube around the estimate. Via-points are represented by black dots. The mean of the trajectories recovered from the demonstrations (without via-points) are depicted by blue lines.

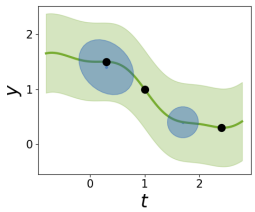


Figure 7: Example of time-driven 1D-trajectory predicted by KMP with three via-points. The reference trajectory of KMP is based on the 2-components GMM of Fig. 3a, represented by blue ellipses. The lengthscale parameter of the Gaussian kernel is fixed as $\sigma_l = 1$. The via-points are represented with black dots.

B Computation Times

Table 1 gives the computation time for one test data with the training data of the real robot experiment of Section 4 with a non-optimized Python code on a laptop with 2.7GHz CPU and 32 GB of RAM.

Table 1: Computation time of GMR, MOGP and GMR-based GP for one test data in the real robot experiment of Section 4. All the time values are given in milliseconds [ms].

GMR	MOGP	GMR-based GP
1 ± 0.1	4 ± 0.6	13 ± 1