

Bayesian Uncertainty Modeling for Programming by Demonstration

Jonas Umlauf¹, Yunis Fanger, Sandra Hirche¹

Abstract—Programming by Demonstration allows to transfer skills from human demonstrators to robotic systems by observation and reproduction. One aspect that is often overlooked is that humans show different trajectories over multiple demonstrations for the same task. Observed movements may be more precise in some phases and more diverse in others. It is well-known that the variability of the execution carries important information about the task. Therefore, we propose a Bayesian approach to model uncertainties from training data and to infer them in regions with sparse information. The approach is validated in simulation, where it shows higher precision than existing methods, and a robotic experiment with variance based impedance adaptation.

I. INTRODUCTION

Transferring skills from humans to autonomous systems, also known as Programming by Demonstration (PbD) [1], [2], is becoming increasingly important as robots are more deployed in complex and unknown environments. For the reproduction of demonstrated trajectories, the representation of desired motions with dynamical systems yields many advantages, e.g., the high robustness to spatial disturbances, the ability to generalize in areas not covered by training data and real-time capability. The proposed methods of Dynamic Movement Primitives (DMPs) [3] and Stable Estimator of Dynamical Systems (SEDS) [4] provide additional convergence guarantees for point-to-point and show promising results.

However, these approaches are missing a proper handling of the uncertainty inherent to training data shown by a human demonstrator. In case of multiple demonstration trajectories for the same task, the variability in the training set is a essential piece of information which must be taken into account during reproduction. For example, in pick and place tasks, humans show high accuracy in the crucial parts at the beginning and the end of the movement while the intermediate motion varies more. Having knowledge about the variability of training data available during reproduction allows to take additional aspects of the task into account, e.g., optimizing execution time or increasing safety through higher compliance. Thus, a flexible representation of variability which precisely models this changing uncertainty is required.

The adaptation of the control decision based on uncertainty in the task description is also an active topic of research. Risk-sensitive control algorithms [5] allow to weight higher-order moments of the cost function to favor/avoid uncertain

areas or adapt feedback gains, i.e., stiffness/damping. The approach in [6] adapts the cooperative behavior of an agent in a team by being less dominant if the uncertainty about the task increases. Knowledge about the variance in training trajectories are also utilized in uncertainty-based impedance adaptation. Hereby the stiffness and damping parameters of the robotic manipulator are adapted online, thus changing the dynamic coupling of externally applied forces and positional displacement. Varying stiffness and damping are particularly useful in the trade off between safety (in case of collision) and tracking performance for which various techniques have been proposed: The approach in [7] follows a sliding window idea, thus the variability of nearby training data serves as uncertainty measure. Since the empirical covariance estimation uses a nearest neighbor lookup table, reproduction is only performed well close to the demonstration data. Thus, the inferred uncertainty is imprecise in unseen areas of the task space and also not continuous which leads to jumping stiffness parameters. Alternatively, the work in [8] employs Gaussian Mixture Models (GMMs) for uncertainty estimation which are also not capable to incorporate prior knowledge to improve generalization in unknown areas.

Therefore, this paper proposes new methods for learning uncertainties in PbD tasks, which overcome the weaknesses of previously mentioned techniques. They are based on Gaussian Process Regression, thus coming with the advantage of a Bayesian background. This allows to precisely control the behavior in regions far away from training data while the appropriate kernel choice implies smoothness of the inferred uncertainties. As experimental validation we apply these new methods of uncertainty modeling to the reproduction of demonstrated motions, and further to impedance adaptation of a robotic manipulator.

The remainder of this paper is structured as follows: After formulating the problem in Section II, Section III reviews fundamentals, namely Gaussian Processes and Wishart Processes. Section IV introduces the proposed methods for uncertainty modeling, followed by their application to PbD in Section V.

II. PROBLEM DESCRIPTION

Given¹ are K demonstration trajectories of length T for the same task. Each input data-point (temporal or spatial) is denoted by $\xi_{k,t} \in \mathcal{X} \subset \mathbb{R}^e$ and the corresponding output is denoted by $y_{k,t} \in \mathbb{R}^d$. It is assumed that the observed outputs

¹ Member of the Chair of Information-oriented Control, Department of Electrical and Computer Engineering, Technical University of Munich, D-80290 Munich, Germany [jonas.umlauft,yunis.fanger,hirche]@tum.de

¹Notation: Little bold symbols denote vectors. Capital bold symbols denote matrices. I_d denotes the $d \times d$ identity matrix. $\mathbf{A} \succ 0$, $|\mathbf{A}|$ denote positive definiteness and the determinant of the matrix \mathbf{A} , respectively. $\mathbb{C}[\mathbf{A}]$ denotes the column-wise covariance.

are subject to randomness, specified by an input dependent normal distribution

$$\mathbf{y}_{k,t} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\xi}_{k,t}), \boldsymbol{\Sigma}(\boldsymbol{\xi}_{k,t})). \quad (1)$$

While many regressions methods exist model the mean function $\boldsymbol{\mu}(\boldsymbol{\xi}_{k,t})$, e.g., [9], the variability of \mathbf{y} is rarely modeled explicitly. Therefore, we focus to model changing variability through the input dependent covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\xi}_{k,t})$. The idea is to learn a mapping from the input space \mathcal{X} to the set of all symmetric positive definite (PDF) matrices

$$\mathbb{S}_+^d = \{\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d} | \boldsymbol{\Sigma} \succ 0, \boldsymbol{\Sigma}^\top = \boldsymbol{\Sigma}\}, \quad (2)$$

denoted by $\boldsymbol{\Sigma}(\boldsymbol{\xi}): \mathcal{X} \rightarrow \mathbb{S}_+^d$. A difficult aspect of modeling uncertainties is the bias-variance trade-off since fast changing outputs can either be explained by a strongly varying mean function $\boldsymbol{\mu}(\boldsymbol{\xi}_{k,t})$ or a large variance $\boldsymbol{\Sigma}$. If multiple observations are made at the same input point, an empirical mean and variance is directly computed and used to train the mappings $\boldsymbol{\mu}(\boldsymbol{\xi}_{k,t}), \boldsymbol{\Sigma}(\boldsymbol{\xi}_{k,t})$ separately. When data-points are not aligned, this is not possible. Therefore, smoothness properties of the mean function are assumed and mean and variance function are jointly trained through likelihood optimization as discussed in more detail later. Important to note is that the approach assumes a unimodal distribution for \mathbf{y} , which makes it not suitable, e.g., when navigating around obstacles on two different paths.

We focus on three aspects for proper uncertainty modeling:

- The function $\boldsymbol{\Sigma}(\boldsymbol{\xi})$ is continuous over its input space \mathcal{X} .
- Prior knowledge is properly incorporated using a Bayesian approach, thus in regions where less training data is given, a prior function takes over

$$\lim_{|\boldsymbol{\xi} - \boldsymbol{\xi}_{k,t}| \rightarrow \infty} \boldsymbol{\Sigma}(\boldsymbol{\xi}) = \boldsymbol{\Sigma}_{\text{prior}}(\boldsymbol{\xi}).$$

- Since the complexity of the movement is not known in advance the model should grow with the number of training points to represent arbitrary complex motions.

III. BACKGROUND

This section introduces Gaussian and Wishart Processes, which we propose for uncertainty modeling in PbD.

A. Gaussian Processes

A Gaussian Process (GP) is a stochastic process often considered as a distribution over functions [10]. It assigns to any finite subset $\{\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_M\} \subset \mathbb{R}^e$ in a continuous input domain a joint Gaussian distribution. The GP is denoted by

$$g(\boldsymbol{\xi}) \sim GP(m(\boldsymbol{\xi}), k(\boldsymbol{\xi}, \boldsymbol{\xi}')) \quad (3)$$

and fully specified by its mean and covariance function $m(\boldsymbol{\xi}), k(\boldsymbol{\xi}, \boldsymbol{\xi}')$. If no prior knowledge on the function g is available the mean function is set to zero. A widely used covariance function is the squared exponential kernel

$$k_{\text{SE}}(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sigma_g^2 \exp\left(-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}')^\top \boldsymbol{\Omega}^{-2}(\boldsymbol{\xi} - \boldsymbol{\xi}')\right). \quad (4)$$

It is also employed here, as it leads to smooth functions g as desired. The parameters $\boldsymbol{\Omega}$, σ_g as well as σ_n , which

denotes the variance of the observation noise, are considered hyperparameters of the GP and are concatenated in the vector $\boldsymbol{\theta}$.

GPs are suitable for solving regression tasks by learning the latent function $g(\boldsymbol{\xi})$ from N training data points $\{\boldsymbol{\xi}_i, y_i\}_{i=1}^N$. Here $y_i = g(\boldsymbol{\xi}_i) + \epsilon_i$ is a noisy observation of the latent function with $\epsilon_i \sim \mathcal{N}(0, \sigma_n)$. For any test point $\boldsymbol{\xi}_*$ the mean and variance of the output random variable $y_* \sim \mathcal{N}(\mu_*, \sigma_*)$ are inferred as follows

$$\mu_* = m(\boldsymbol{\xi}_*) + k_{\boldsymbol{\Xi}, \boldsymbol{\xi}_*}^\top (k_{\boldsymbol{\Xi}, \boldsymbol{\Xi}} + \sigma_n^2 \mathbf{I}_N)^{-1} \mathbf{y}, \quad (5)$$

$$\sigma_* = k(\boldsymbol{\xi}_*, \boldsymbol{\xi}_*) - k_{\boldsymbol{\Xi}, \boldsymbol{\xi}_*}^\top (k_{\boldsymbol{\Xi}, \boldsymbol{\Xi}} + \sigma_n^2 \mathbf{I}_N)^{-1} k_{\boldsymbol{\Xi}, \boldsymbol{\xi}_*}, \quad (6)$$

where $\boldsymbol{\Xi} \in \mathbb{R}^{e \times N}$, $\mathbf{y} \in \mathbb{R}^N$ concatenate all $\boldsymbol{\xi}_i, y_i$, respectively and $k_{\boldsymbol{\Xi}, \boldsymbol{\Xi}}, k_{\boldsymbol{\Xi}, \boldsymbol{\xi}_*}$ denotes the matrix/vector of pairwise kernel evaluations. For multiple output dimensions, multiple independent GPs are employed with same input, but differing output data sets.

B. Wishart Processes

To model uncertainties by means of covariance matrices, we review the Wishart distribution $\mathcal{W}(\mathbf{V}, \nu)$, a distribution over matrices $\boldsymbol{\Sigma} \in \mathbb{S}_+^d$. It is constructed from the sum of ν outer products of multivariate Gaussian random variables with covariance $\mathbf{V} \in \mathbb{S}_+$

$$\boldsymbol{\Sigma} = \sum_{i=1}^{\nu} \mathbf{u}_i \mathbf{u}_i^\top \sim \mathcal{W}(\mathbf{V}, \nu), \quad \text{with } \mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{V}). \quad (7)$$

Parameters are the scale matrix \mathbf{V} and the degree of freedom $\nu \in \mathbb{N}, \nu > d - 1$. Its probability density function is

$$p(\boldsymbol{\Sigma} | \mathbf{V}, \nu) = \frac{|\boldsymbol{\Sigma}|^{(\nu-d-1)/2}}{2^{\nu d/2} |\mathbf{V}|^{\nu/2} \Gamma_d(\nu/2)} \exp\left(-\frac{1}{2} \text{Tr}(\mathbf{V}^{-1} \boldsymbol{\Sigma})\right),$$

where $\Gamma_d(\cdot)$ is the multivariate gamma function. The Wishart distribution is the multivariate generalization of the Chi-square distribution for $V = d = 1$.

Similarly to how the Gaussian distribution is generalized to Wishart Processes, one can extend Wishart distributions to Wishart Processes (WPs), which are introduced in the following according to [11]. The WP is obtained by replacing the Gaussian random variables \mathbf{u}_i by νd Gaussian processes, thus $u_{i,j}(\boldsymbol{\xi}) \sim \mathcal{GP}(0, k(\boldsymbol{\xi}, \boldsymbol{\xi}'))$, where $i = 1, \dots, \nu$ and $j = 1, \dots, d$ with $\mathbf{u}_i(\boldsymbol{\xi}) = [u_{i,1}(\boldsymbol{\xi}), \dots, u_{i,d}(\boldsymbol{\xi})]^\top$

$$\boldsymbol{\Sigma}_{\text{WP}}(\boldsymbol{\xi}) = \sum_{i=1}^{\nu} \mathbf{L} \mathbf{u}_i(\boldsymbol{\xi}) \mathbf{u}_i(\boldsymbol{\xi})^\top \mathbf{L}^\top \sim \mathcal{WP}(\mathbf{V}, \nu), \quad (8)$$

where $\mathbf{L} \mathbf{L}^\top = \mathbf{V}$ is the lower Cholesky decomposition. Thus each element in the input space $\boldsymbol{\xi}$ is assigned a distribution over a positive definite matrix. In this application case, we are not interested in the full distribution over PDF matrices and only consider the GP mean. Therefore, given a test input $\boldsymbol{\xi}^*$ the inference method first evaluates the mean prediction of the νD GPs at $\boldsymbol{\xi}^*$ according to (5), which we denote by \mathbf{u}_i^* . Substituting this in (8) provides the inferred PDF matrix $\boldsymbol{\Sigma}^*$.

IV. MODELING UNCERTAINTIES

A. Training Wishart Processes

To model input dependent uncertainties, we learn a mapping from the input space \mathcal{X} (task space, time, etc) to covariance matrices in \mathbb{S}_+ . To learn such a mapping from training data consider the set of observations $\mathcal{D} = \{\boldsymbol{\xi}_{k,t}, \mathbf{y}_{k,t}\}_{k=1,\dots,K}^{t=1,\dots,T}$, with $\boldsymbol{\xi}_{k,t} \in \mathbb{R}^e$ and $\mathbf{y}_{k,t} \in \mathbb{R}^d$ concatenated in Ξ and \mathbf{Y} , respectively. The training of the Wishart Process is an optimization over the parameters $\mathbf{U}, \mathbf{L}, \Theta$. The matrix $\mathbf{U} \in \mathbb{R}^{\nu d \times TK}$ concatenates the outputs of the νd GPs which cannot be observed directly. The lower triangular matrix \mathbf{L} contains $d(d+1)/2$ free elements and is initialized to $\mathbf{L} = \mathbf{I}_d$. The matrix Θ denotes the concatenation $\Theta = [\boldsymbol{\theta}_{1,1}, \dots, \boldsymbol{\theta}_{\nu,d}]$ of the hyperparameters of the νd GPs. To avoid a mixed integer optimization problem, we do not optimize over ν but set $\nu = d + 1$ as proposed in [11]. In the following, we propose the use of two different cost functions for the optimization over those parameters depending on the alignment of the training data.

a) Mean squared error (MSE): This method is only applicable for aligned training data because if multiple observations are made at the same input, the empirical mean and variance at this point is used for training. As cost function for learning the variance mapping, we use the mean squared error between the empirical and the inferred covariances at all training points. Required are training data-points with the same input locations across the K trajectories, thus $\boldsymbol{\xi}_t := \boldsymbol{\xi}_{1,t} = \boldsymbol{\xi}_{2,t}, \dots, = \boldsymbol{\xi}_{K,t} \forall t$. This assumption is often fulfilled whenever $\boldsymbol{\xi}$ is a one dimensional (temporal) variable and data is recorded at a uniform sampling rate. It allows to compute a training set for the uncertainty $\{\boldsymbol{\xi}_t, \boldsymbol{\Sigma}_t\}_{t=1}^T$, where $\boldsymbol{\Sigma}_t = \mathbb{C}[\mathbf{Y}_t]$ is the empirical covariance with $\mathbf{Y}_t = [\mathbf{y}_{1,t}, \mathbf{y}_{2,t}, \dots, \mathbf{y}_{K,t}]$. The optimization over the MSE between the empirical covariances $\boldsymbol{\Sigma}_t$ and the inferred matrices from the Wishart Process $\boldsymbol{\Sigma}_{\text{WP}}(\boldsymbol{\xi}_t)$ is computed as follows

$$\min_{\mathbf{U}, \mathbf{L}, \Theta} J_{\text{MSE}} = \min_{\mathbf{U}, \mathbf{L}, \Theta} \sum_{t=1}^T \frac{1}{T} \|\boldsymbol{\Sigma}_t - \boldsymbol{\Sigma}_{\text{WP}}(\boldsymbol{\xi}_t)\|^2. \quad (9)$$

For computational efficiency, elements of the Cholesky decomposition of $\boldsymbol{\Sigma}_t, \boldsymbol{\Sigma}_{\text{WP}}$ are vectorized and the Euclidean norm of the difference of these two vectors is taken. The GP outputs \mathbf{U} only contain $\nu d \times T$ elements, which simplifies the optimization problem. This estimation is only possible if the empirical covariances can be computed (with aligned training data). If this is not the case a maximum likelihood cost function must be employed as presented in the following.

b) Maximum Likelihood: Since each training point is generated from a Gaussian distribution with varying mean and variance as given in (1) the computation of the likelihood requires to infer the mean and the covariance simultaneously. For the mean prediction, the mean of d GPs are taken $\boldsymbol{\mu}_{\text{GP}}(\boldsymbol{\xi}_{k,t})$ as introduced in (5). The GPs are trained using the data set $\boldsymbol{\xi}_{k,t}, \mathbf{y}_{k,t}$ by default hyperparameter optimization.

To infer the covariance matrices, we employ Wishart Processes as defined in (8), and maximize the likelihood

given by

$$J_{\text{like}} = \prod_{t,k=1}^{T,K} \mathcal{N}(\mathbf{y}_{k,t} | \boldsymbol{\mu}_{\text{GP}}(\boldsymbol{\xi}_{k,t}), \boldsymbol{\Sigma}_{\text{WP}}(\boldsymbol{\xi}_{k,t})) \quad (10)$$

$$= \prod_{t,k=1}^{T,K} |2\pi \boldsymbol{\Sigma}_{\text{WP}}(\boldsymbol{\xi}_{k,t})|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \mathbf{w}_{k,t}^\top \boldsymbol{\Sigma}_{\text{WP}}(\boldsymbol{\xi}_{k,t})^{-1} \mathbf{w}_{k,t}\right),$$

where $\mathbf{w}_{k,t} = \mathbf{y}_{k,t} - \boldsymbol{\mu}_{\text{GP}}(\boldsymbol{\xi}_{k,t})$. For numerical stability the negative log likelihood is minimized instead

$$\min_{\mathbf{U}, \mathbf{L}, \Theta} -\log J_{\text{like}}. \quad (11)$$

B. Prior uncertainty

As WPs inherit they Bayesian setup from GPs, the behavior far away from training points can be set by injecting prior knowledge. By setting a nonzero prior mean function $m_{i,j}(\boldsymbol{\xi})$ for the GPs $\mathbf{u}_i(\boldsymbol{\xi}) \sim \mathcal{GP}$ the inferred covariance of the WP is properly defined in areas without training data. For instance, if we assume large uncertainty in areas far away from training points, the prior is set to large values resulting in higher inferred covariance matrices outside the training area. The prior mean function can be set to a constant value or to an arbitrarily complex smooth function.

C. Distinction between two uncertainty types

In PbD, one can distinguish between uncertainty due to

- missing data in regions of the input space (Type I) or
- contradiction/high variability of training data (Type II).

For specific applications, Type I uncertainty might require a different response by the robot than Type II uncertainty. For example Type I might cause the robot to wait at the current locations to query for more training data and Type II is supposed to result in low stiffness of the end-effector.

The variance provided, e.g., by GPs in (6) is independent from the outputs \mathbf{y} and therefore only provides a measure for the proximity to the training points in the input space, which corresponds to Type I uncertainty. On the other hand, Gaussian Mixture Models (GMMs) only provide Type II uncertainty. With the use of WPs, we propose a Bayesian approach which is well suited to model both types simultaneously. It is achieved by modeling the uncertainty explicitly rather than considering it as a error term or noise in the mean regression. By choosing zero prior mean function and utilizing (6) our model still contains both types separately. In this work, we do not further distinguish and only use one model for the joint representation of both uncertainties.

D. Approach to Learning the Cholesky Decomposition

As the Wishart Process approach requires the optimization of a large number of parameters, we propose an alternative method in this section, which constructs the PDF matrix not from an outer product of vectors but from a Cholesky decomposition. It is known that every PDF matrix $\boldsymbol{\Sigma} \in \mathbb{S}_+^d$ can be represented by a lower triangular matrix \mathbf{L}_{CD} such that $\boldsymbol{\Sigma} = \mathbf{L}_{\text{CD}} \mathbf{L}_{\text{CD}}^\top$ which has $d(d+1)/2$ nonzero elements. We therefore vectorize \mathbf{L}_{CD} in $\mathbf{l} \in \mathcal{L} \subset \mathbb{R}^{d(d+1)/2}$. So rather

than constraining the output of the learned function $\mathcal{X} \rightarrow \mathbb{R}^{d \times d}$ to the manifold of PDF matrices, we propose to learn $\mathcal{X} \rightarrow \mathbb{R}^{d(d+1)/2}$. Thus, by using the Cholesky decomposition approach, the number of optimization parameters is reduced considerably. To stick to a Bayesian framework, we learn this mapping using GPs. As with WPs, we distinguish between the two different cases how training data is available:

a) *Aligned input location across all trajectories*: Again, this allows to compute a training set for the uncertainty $\{\xi_t, \Sigma_t\}_{t=1}^T$. Applying Cholesky decomposition to the empirical covariances Σ_t transforms this into $\{\xi_t, \mathbf{l}_t\}_{t=1}^T$, with $\mathbf{l} \in \mathcal{L} \subset \mathbb{R}^{d(d+1)/2}$ and $d(d+1)/2$ Gaussian Processes are directly trained with likelihood optimization. Here the GP outputs are directly observable and we only consider the mean function of the GP since a distribution of triangular matrices with normal distributed entries has only been studied very little [12].

b) *Arbitrarily located training data*: As discussed above for this more general case, the empirical covariances can not be determined. But analogously to the WPs, the likelihood in (10), can still be evaluated. Therefore, the optimization problem becomes

$$\min_{\mathbf{U}_{\text{CD}}, \Theta} -\log \prod_{t,k=1}^{T,K} \mathcal{N}(\mathbf{y}_{k,t} | \boldsymbol{\mu}_{\text{GP}}(\xi_{k,t}), \boldsymbol{\Sigma}_{\text{CD}}(\xi_{k,t})), \quad (12)$$

where $\mathbf{U}_{\text{CD}} \in \mathbb{R}^{TK \times d(d+1)/2}$ is the concatenation of all unknown GP outputs $\mathbf{l}_{k,t} \in \mathbb{R}^{d(d+1)/2}$, and *CD* labels the Cholesky decomposition.

c) *Inference*: Given a test point ξ_* , each entry of \mathbf{l} is inferred according to the mean of the corresponding GP as in (5). Hereby the $(d+1)d/2$ inferred values $\mathbf{l}_{*,j}$ are the entries of the lower triangular matrix $\mathbf{L}_{\text{CD}*}$. The inferred PDF matrix is obtained from

$$\boldsymbol{\Sigma}_{\text{CD}} = \boldsymbol{\Sigma}(\xi_*) = \mathbf{L}_{\text{CD}*} \mathbf{L}_{\text{CD}*}^{\text{T}}. \quad (13)$$

In the same way as prior knowledge about the uncertainty was injected to WPs, the zero mean priors for the GPs can be set to nonzero values in the Cholesky decomposition approach.

E. Simulation

For demonstration purposes, we provide an example for one input ($e = 1$) and two output ($d = 2$) dimensions. The training data is generated from

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}(\xi), \boldsymbol{\Sigma}(\xi)), \quad \xi \in [-100, 100], \quad (14)$$

$$\boldsymbol{\mu}(\xi) = \begin{bmatrix} 30 \sin\left(\frac{\xi}{20}\right) \\ 30 \tanh\left(\frac{\xi}{20}\right) \end{bmatrix},$$

$$\boldsymbol{\Sigma}(\xi) = \begin{bmatrix} 150 \exp\left(\frac{-\xi}{50}\right)^2 & 0 \\ 0 & \xi + 150 \end{bmatrix},$$

with $T = 10$ (ξ_t are 10 equally spaced values in $[-100, 100]$) and $K = 50$ (50 draws for \mathbf{y} at one ξ_t). We run the Wishart Process Section IV-A and the Cholesky decomposition Section IV-D approach for comparison. For WP

Method	CD	WP	GMM $\mathcal{K} = 8$	GMM $\mathcal{K} = 5$
MSE	3.54e-13	0.0206	1.05	7.71
Time [s]	0.167	8.64	0.515	0.237

TABLE I

COMPARISON OF THE MEAN SQUARE ERROR FOR WP, CD AND GMM.

we set, as mentioned previously, $\nu = d + 1 = 3$ and the νd GP mean priors are all set to $m(\xi) = 8$. For the CD approach, we set the priors of the diagonal elements to one, thus $m(\xi) = 1$ and all others to zero. As both proposed cost functions are applicable for this data set, we run them for comparison as well. In order to compare with previously existing approaches, we evaluate GMMs with $\mathcal{K} = 5$ components on the same dataset.

a) *Results*: The optimization using MSE and likelihood cost function yield similar results in terms of precision and speed. We therefore used J_{like} in the further simulation as it is more general applicable. For data at same input locations, the CD approach requires only to optimize the hyperparameters of the GP, therefore there was no need to employ (12). The results for the all three methods are shown in Fig. 1 to Fig. 3. Table I shows the mean squared error (MSE) between inferred and empirical covariances (not the reference in (14)) and the optimization time (on a standard laptop in Matlab) for all methods. In addition it shows the GMM for $\mathcal{K} = 8$ as it is the best of all $\mathcal{K} = 1 \dots 10$, but fails in the prediction of the mean, therefore the plot for $\mathcal{K} = 5$ is shown.

b) *Discussion*: Table I shows that the Cholesky decomposition approach outperforms other methods in speed and precision, GMMs show the lowest precision. From Fig. 1 to Fig. 3 it can be observed that the inferred covariances for WPs and CD increase in regions without training data, while GMR provides a "nearest neighbor" prediction. This reflects that our approaches allow to include meaningful prior knowledge. In addition, the GMMs suffer from the parametric design which makes it sensitive to the choice of the number of mixtures. Our non-parametric design allows to model arbitrary complex mappings with (in case of CD) a fixed number of hyperparameters. Since all three approaches require nonconvex optimization, performance guarantees cannot be given for any of them. Yet the small MSE suggests that a reasonable small minimum was found.

Comparing Wishart Processes and the Cholesky decomposition shows that the required number of GPs for the WP is νd which (in principle grows linearly) with the number of output dimensions d . But since ν is chosen (in accordance to [11]) equal to the number of output dimensions $d + 1$, it increases quadratically. The approach using the Cholesky decomposition requires $d(d+1)/2$ GPs, thus half of the WPs but also increases quadratically in d . Another advantage of the Cholesky method is (in case of aligned data) the exact output values for the GPs given by $\boldsymbol{\Sigma}$, rather than optimizing over those how it is done with WPs. When data is distributed arbitrarily both require similar computational effort for solving (12) or (11). One major advantage of using WPs is their well founded probabilistic theoretical

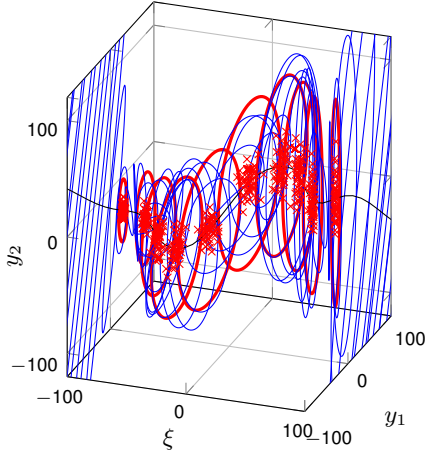


Fig. 1. Wishart Process approach

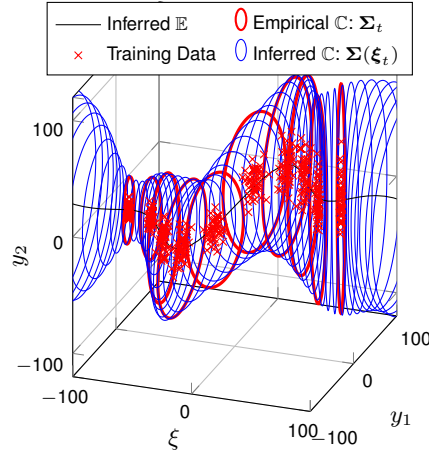


Fig. 2. Cholesky decomposition approach

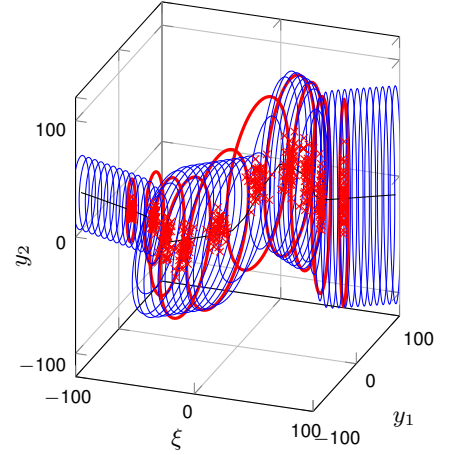


Fig. 3. GMM with 5 Gaussians

background as Wishart distributions are widely applied in Bayesian modeling. In contrast, the construction of PDF matrices from triangular matrices with normal distributed entries is less common, therefore the theoretical foundation has not been investigated in depth. However, we only use the deterministic perspective of the GP and therefore a probabilistic analysis is not inevitable. For further experiments, we consider CD for variance prediction due to its higher precision.

V. ROBOTIC EXPERIMENT

One possible application for the uncertainty measure in robotics is the proper trade-off between performance and safety for which the manipulator's impedance is adapted. The end-effector is governed by the impedance control law

$$\mathbf{f} = \mathbf{K}(\mathbf{x}_d - \mathbf{x}) + \mathbf{D}(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}), \quad (15)$$

where $\mathbf{f} \in \mathbb{R}^d$, $\mathbf{x}, \mathbf{x}_d \in \mathbb{R}^d$ and $\mathbf{K}, \mathbf{D} \in \mathbb{S}_+^d$ denote the end-effector force, the (desired) position in the task space and the stiffness/damping matrices, respectively. The impedance \mathbf{K}, \mathbf{D} is adapted online based on the uncertainty. For the validation of the presented methods in a real-world PbD task, we conduct an experiment which employs the proposed uncertainty modeling for impedance adaptation.

A. Setup

As a robotic manipulator, a 2 degree of freedom (DoF) linear axis device with mounted 6 DoF JR3 force-torque sensor was used. The controller is implemented in Matlab/Simulink running on a Linux real-time kernel and an update rate of 1kHz. To define the task, a 50 inch monitor displays the virtual environment and the manipulators position in real-time. A picture of the hardware setup is shown in Fig. 4. The experiment itself is separated in three phases:

Demonstration: A human operator provides $K = 6$ training trajectories of the same task. The task is to move from a start to a goal position through an environment which is wide open at the start but constrained through a narrow tunnel towards

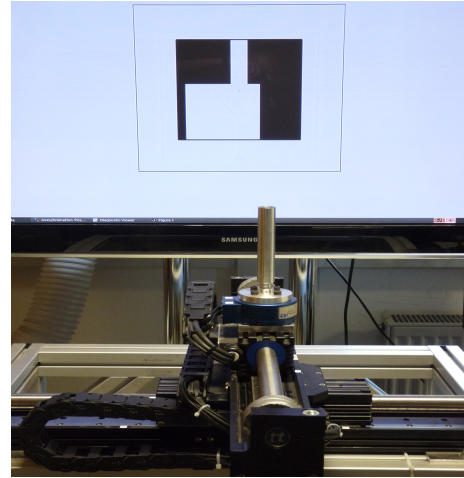


Fig. 4. 2DoF Linear axes setup with force-torque sensor and display showing the virtual environment.

the end. The environment is constant across all trials, but it allows different paths to be chosen from the start to the goal point as shown in Fig. 6. The movement in x -direction is free while in y -direction a constant velocity is predetermined.

Learning: The trajectory of the generalized motion is encoded using a Gaussian Processes and Dynamic Movement Primitives (DMPs) as introduced in [6]. More importantly, a model for the time dependent variability is learned from data as described in Section IV. The required optimization (12) is solved offline.

Reproduction: The robotic manipulator uses DMPs to follow the demonstrated trajectory online, while the admittance controller (15) adapts the stiffness and damping online [7], depending on the inferred uncertainties from the CD-model. To verify changing admittance during the task, manual disturbances are applied during execution.

B. Dynamic Movement Primitives

The generation of trajectories imitating an observed demonstration is achieved using DMPs which are capable

of representing complex nonlinear movements, while guaranteeing stable goal converging behavior [3]. They are based on a set of differential equations: The *point attractive system*

$$\tau \dot{z} = \alpha(\beta(\mathbf{g} - \mathbf{x}) - z) + \mathbf{f}^{\text{DMP}}(s), \quad \tau \dot{\mathbf{x}} = z, \quad (16)$$

consists for goal directed movements of a linear part which pulls the position \mathbf{x} towards the attractor at the goal point \mathbf{g} and a nonlinearity $\mathbf{f}^{\text{DMP}}(s)$ which encodes the trained motion. The variables $\alpha, \beta > 0$ determine goal convergence of the linear part while $\tau > 0$ adapts the speed of reproduction. The *canonical system*

$$\tau \dot{s} = -\gamma s, \quad \gamma > 0, \quad s(0) = s_0 > 0 \quad (17)$$

regulates evolution of the phase variable s and therefore the speed of $\mathbf{f}^{\text{DMP}}(s)$. This nonlinearity is learned using GP Regression on the training data $\{s_{k,t}, \mathbf{f}_{k,t}^{\text{DMP}}\}_{k=1,\dots,K}^{t=1,\dots,T}$ for K demonstrations of length T as described in [6]. This training data set is determined by simulating $s_{k,t}$ from (17) and computing $\mathbf{f}_{k,t}^{\text{DMP}}$ from

$$\mathbf{f}_{k,t}^{\text{DMP}} = \tau^2 \ddot{\mathbf{x}}_{k,t} - \alpha(\beta(\mathbf{g} - \mathbf{x}_{k,t}) - \tau \dot{\mathbf{x}}_{k,t}), \quad \forall k, t, \quad (18)$$

where the position $\mathbf{x}_{k,t}$ is observed and velocity $\dot{\mathbf{x}}_{k,t}$ and acceleration $\ddot{\mathbf{x}}_{k,t}$ are computed numerically. To match the notation introduced above, the phase variable s corresponds to the one dimensional input ξ and the d dimensional output \mathbf{f}^{DMP} corresponds to \mathbf{y} .

C. Stiffness Adaptation

For modifying the stiffness based on the variability in the data, our proposed method is now applied to the training data set $\{s_{k,t}, \mathbf{f}_{k,t}^{\text{DMP}}\}_{k=1,\dots,K}^{t=1,\dots,T}$ which guides the DMP. Therefore, we obtain the same scenario as in the simulation with 1D input and 2D output. Since the training data is sampled at a constant rate the input is aligned. For every input s which is generated during run-time, the CD method infers a positive definite matrix which reflects the variability at this part of the task during the demonstration. This is now used for stiffness adaptation as following:

- Since high variability indicates sufficient freedom in the environment, we allow the manipulator to deviate further from the mean of demonstrations which is encoded in a lower stiffness. This allows for higher safety level in the presence of humans in case of unintended collision. It also allows the robot to be actively guided by an expert during specific phases of the task.
- On the other side, low variability in the demonstration here indicates tight spatial constraints. Therefore, high precision in tracking the demonstrations is required. To ensure this also in the presence of external perturbation, the stiffness is increased.

While our method for deriving the covariance matrices differs from [7], the subsequent computation of the stiffness and damping matrices \mathbf{K} , \mathbf{D} is the same. First, the eigenvalue decomposition $\Sigma = \mathbf{Q}\Lambda\mathbf{Q}^\top$ of the positive definite covariance matrix is performed where \mathbf{Q} contains the Eigenvectors and Λ is the matrix with the Eigenvalues $\lambda_i, i \in [1, 2, \dots, d]$ on

α, β, τ	γ	$\underline{\sigma}$	$\bar{\sigma}$	\underline{k} [N/m]	\bar{k} [N/m]
1	0.7	$3.5e-4$	$1.2e-3$	50	1000

TABLE II

PARAMETERS USED IN THE EXPERIMENT.

its diagonal. By taking the square-root of these Eigenvalues the variances in direction of the Eigenvectors are obtained $\sigma_i = \sqrt{\lambda_i}$. The stiffness matrix is then constructed as $\mathbf{K} = \mathbf{Q}\Gamma\mathbf{Q}^\top$ with

$$\Gamma_{i,j} = \begin{cases} \gamma(\sigma_i) & i = j \\ 0 & i \neq j \end{cases}, \quad (19)$$

where

$$\gamma(\sigma_i) = \begin{cases} \frac{\underline{k}}{\bar{k}} & \bar{\sigma} \leq \sigma_i \\ \frac{\underline{k}}{\bar{k}} - (\bar{k} - \underline{k}) \frac{\sigma_i - \underline{\sigma}}{\bar{\sigma} - \underline{\sigma}} & \underline{\sigma} \leq \sigma_i \leq \bar{\sigma} \\ \frac{\underline{k}}{\bar{k}} & \sigma_i \leq \underline{\sigma} \end{cases}. \quad (20)$$

The sensitivity to variance and bounds for the stiffness are determined by the parameters $\bar{\sigma}, \underline{\sigma}, \bar{k}, \underline{k}$. The damping matrix is similarly determined as $\mathbf{D} = \mathbf{Q}\bar{\Gamma}\mathbf{Q}^\top$ with $\bar{\Gamma}_{i,j} = \sqrt{\Gamma_{i,j}}$. All parameters chosen for the experiment are shown in Table II. A diagram of the entire learning and control scheme is shown in Fig. 5.

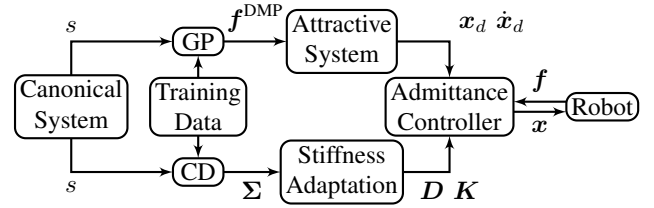


Fig. 5. The overall design of the framework

D. Results

The experimental environment with the recorded training trajectories and reproduction is shown in Fig. 6. Due to the constraints of the virtual environment, the variability of demonstrations is large at the beginning and decreases to a minimum towards the end. This is also captured by the CD model shown in Fig. 7. Since the movement speed in y is constant across all trajectories, the respective stretch of the ellipses in y direction is minimal, thus leading to constant high stiffness in that direction. Since the x direction is more relevant, Fig. 8 shows the changing stiffness and the effect of manual perturbations. It illustrates, that in regions where demonstrations are less variable, the stiffness increases and thus the positional displacement with comparable perturbation force decreases.

E. Discussion

The results show proper modeling of variability, here as a function of time, and that the proposed methods are applicable on real-world training data. The stiffness adaptation is a relevant application especially in human-robot interaction tasks. Even though the application here uses a time-based approach, our methods are equivalently applicable for input spaces with multiple dimension, e.g., joint space.

VI. CONCLUSION

In this paper, we propose two methods for modeling uncertainties in Programming by Demonstration tasks. It focuses on modeling changing variability of trajectories over multiple demonstrations in a Bayesian setting. Therefore, it allows to inject prior knowledge about uncertainties to which the model returns in areas without training data. As both methods are based on nonparametric Gaussian Process modeling, the flexibility of the model is not fixed by a finite parameter set, but can be arbitrarily complex. We show in simulation the superiority in terms of precision compared to widely used GMMs. In an experiment, we apply our methods to impedance adaptation of a manipulator based on the inferred uncertainties. It shows successful reproduction of the task learned from multiple human demonstrations.

ACKNOWLEDGMENTS

The European Research Council Starting Grant "Control based on Human Models (conhumo)" supported this work under grant agreement number 337654. Published with ©2018 IEEE with DOI 10.1109/ICRA.2017.7989759 [13]

REFERENCES

- [1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Robot programming by demonstration*. Springer, 2008.
- [2] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *International Conference on Machine Learning*, vol. 97, 1997.
- [3] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*. Springer, 2006, pp. 261–280.
- [4] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [5] J. R. Medina and S. Hirche, "Uncertainty-dependent optimal control for robot control considering high-order cost statistics," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3995–4002.
- [6] Y. Fanger, J. Umlauf, and S. Hirche, "Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation," in *International Conference on Intelligent Robots and Systems (IROS)*. Deajeon, Korea: IEEE, Oct. 2016, pp. 3913–3919.
- [7] K. Kronander and A. Billard, "Online learning of varying stiffness through physical human-robot interaction," in *International Conference on Robotics and Automation (ICRA)*. IEEE, May 2012.
- [8] S. M. Khansari-Zadeh, K. Kronander, and A. Billard, "Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control," in *Proceedings of Robotics: Science and Systems*, USA, July 2014.
- [9] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional space," in *International Conference on Machine Learning (ICML)*, 2000, pp. 1079–1086.
- [10] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, Jan. 2006.
- [11] A. Wilson and Z. Ghahramani, "Generalised Wishart processes," in *Conference on Uncertainty in Artificial Intelligence*. Corvallis, Oregon: AUAI Press, 2011, pp. 736–744.
- [12] D. Viswanath and L. Trefethen, "Condition numbers of random triangular matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 19, no. 2, pp. 564–581, 1998.
- [13] J. Umlauf, Y. Fanger, and S. Hirche, "Bayesian uncertainty modeling for programming by demonstration," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 6428–6434.

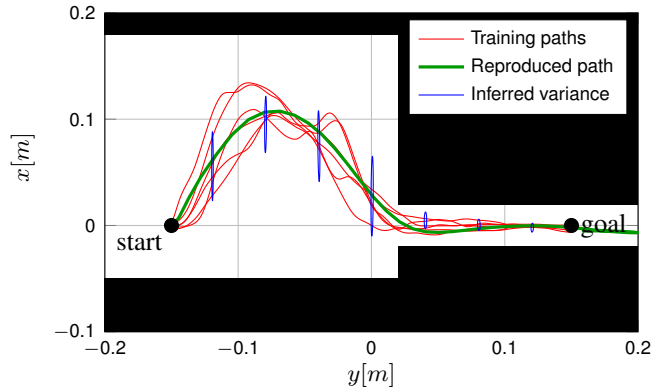


Fig. 6. Shown are six training trajectories provided by a human operator in the virtual environment along with the reproduced trajectory and the inferred variability as ellipsoids.

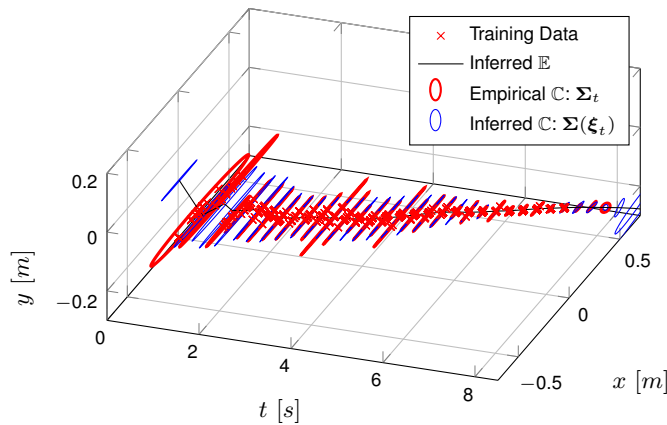


Fig. 7. Empirical and inferred covariances during movement reproduction are shown along with training data and the mean trajectory.

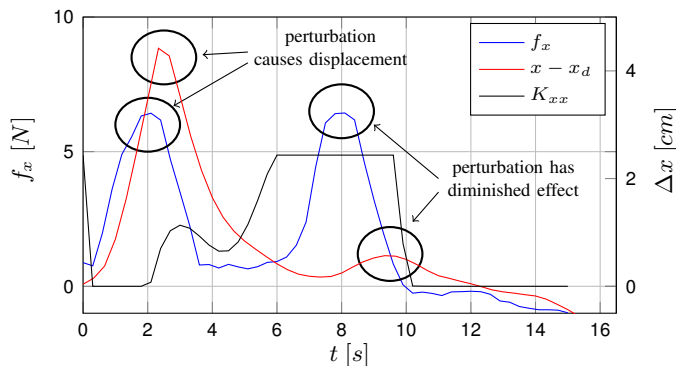


Fig. 8. Deviation from the desired trajectory $\Delta x = x - x_d$, externally applied forces in x direction f_x and stiffness K_{xx} over time in direction during reproduction with two manual perturbations at $t_1 \approx 2$ s and $t_2 \approx 8$ s.