Sparse Gaussian Process Regression for Compliant, Real-Time Robot Control

Jens Schreiter[1], Peter Englert[2], Duy Nguyen-Tuong[1], Marc Toussaint[2]

***Abstract*— Sparse Gaussian process (GP) models provide an efficient way to perform regression on large data sets. The key idea is to select a representative subset of the available training data, which induces the sparse GP model approximation. In the past, a variety of selection criteria for GP approximation have been proposed, but they either lack accuracy or suffer from high computational costs. In this paper, we introduce a novel and straightforward criterion for successive selection of training points used for GP model approximation. The proposed algorithm allows a fast and efficient selection of training points, while being competitive in learning performance. As evaluation, we employ our approach in learning inverse dynamics models for robot control using very large data sets (e.g. 500.000 samples). It is demonstrated in experiments that our approximated GP model is sufficiently fast for real-time prediction in robot control. Comparisons with other state-of-the-art approximation techniques show that our proposed approach is significantly faster, while being competitive to generalization accuracy.**

I. Introduction

Gaussian process regression (GPR) is a powerful and widely used non-parametric Bayesian modeling technique [1] which increasingly becomes interesting for many robotic applications, such as robot control [2], [3]. In contrast to other model learning approaches, such as support vector machines, GPR offers a flexible probabilistic framework leading to predictive distributions for test points, where the model training is easy to achieve with standard Bayesian procedures [1]. However, the applicability of standard GPR to large-scale problems with a high number of training points n is limited, due to the unfavourable scaling in training time and memory requirements. The dominating factors are usually $O(n^3)$ costs for inversion of a dense covariance matrix between all available training points and the $O(n^2)$ space required to store it in memory. Furthermore, the full GPR model needs $O(n)$ costs for the prediction of a test instance. In the past, there have been efforts to alleviate these limitations, enabling GPR learning for large-scale data (see Section II for a brief overview).

To the best of our knowledge, the most existing GPR approximation methods generally have higher computational costs than randomized selection, i.e. random subset selection of available training data, but in exchange yield significantly better results, as random selection typically leads to over- and underfitting. In this paper, we propose a straightforward selection approach which is close to the level of randomized selection in learning speed, but without huge loss in the accuracy. It turns out that our selection method is closely related to the inclusion technique by [4]. However, we are able to further reduce the computational costs significantly, while being competitive in the model accuracy, thanks to several additional reasonable assumptions. For evaluation, we learn inverse dynamics models from large-scale sampled data, while comparing different approximation techniques. Subsequently, our learned models are employed for real-time, compliant control in robot tracking control tasks. Learning inverse dynamics models from data has been shown to be a useful alternative in model-based robot control [2], [5]. This control approach is especially useful in cases where sufficiently precise physical models are hard to obtain analytically. As learning inverse dynamics models requires massive amount of data, our proposed sparse GP approximation technique is able to demonstrate its strengths and advantages.

The remainder of the paper is organized as follows: we provide a brief overview on existing GPR approximation techniques in Section II. Subsequently, we introduce the sparse GPR setting and explain our novel strategy for fast greedy data selection in Section III. We also describe an efficient way to learn the hyperparameters in our specific setup in Section III-C, where a generalized expectation maximization (EM) algorithm is employed. In Section IV, we report on the results of our comprehensive comparison on learning inverse dynamics models. Furthermore, we demonstrate the real-time applicability of learned inverse dynamics models for compliant robot control on a PR2 robot, see Fig. 1. Finally, we conclude and discuss the results of our method in Section V.

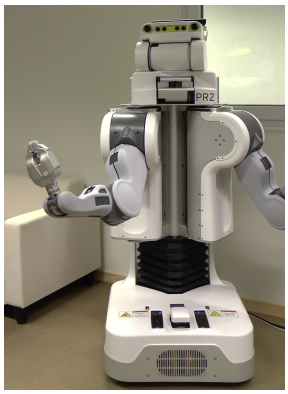


Fig. 1. The PR2 for evaluations.

II. Related Work

To overcome computational limitations of GPR described previously, many approximations for full GPR have been proposed. For example, covariance matrix approximations, such as the Nystrøm method proposed by [6], can be employed to increase the GP learning performance. Also, various sparse likelihood approximations have emerged recently, whose relations have been formalized in a unifying framework [7].

[1]Robert Bosch GmbH, Corporate Research, Department for Cognitive Systems, 70442 Stuttgart, Germany.
Email: jens.schreiter@de.bosch.com
[2]University of Stuttgart, Institute for Parallel and Distributed Systems, Machine Learning and Robotics Laboratory, 70569 Stuttgart, Germany.
Email: peter.englert@ipvs.uni-stuttgart.de

The core idea of the fully independent training conditional (FITC) approximation [8] is to use a flexible subset of virtual training points to generate a sparse GPR model and optimize the virtual training points along with all other hyperparameters. In contrast, the deterministic training conditional (DTC) approximation selects a representative subset of real training points, the so-called active points, which induce a sparse approximation. The approach proposed in [9] employs a fast information gain criterion for insertion of input points. Csató and Opper [10] measure the projection-induced error in the reproducing kernel Hilbert space (RKHS) to choose the representative subset. The authors of [4] use a computational costly selection heuristic which approximate the log marginal posterior probability. All of these DTC approximation methods have higher computational costs than randomized selection, but yield significantly better results.

In robotics domain, [5] proposes a local GPR approach to speed up the learning process. Although the local GPR approach does not belong to typical GP approximation techniques, it enables a fast GPR training appropriate for applications in real-time online learning [5]. However, the focus there is mainly on learning speed while eventually sacrificing model accuracy. In this paper, we take the middle course between high model accuracy and fast computational speed. Our aim is to accelerate the GP approximation process, while staying close to the model quality as demonstrated by state-of-the-art approximation techniques.

III. Efficient Active Set Selection for Sparse Gaussian Process Regression

In this section, we describe the setting of sparse GPR and introduce our criterion for selecting a subset of training points, i.e. the active set. Furthermore, we describe an efficient incremental learning procedure for the GP hyperparameters arising from this approximation.

A. Sparse Gaussian Process Regression

Let $(\boldsymbol{y},\boldsymbol{X})$ be the training data set, where $\boldsymbol{y} \in \mathbb{R}^n$ is a vector of noisy observations of the underlying scalar function $f(\boldsymbol{x}_i) = f_i$, obeying the relationship $y_i = f_i + \varepsilon_i$ with Gaussian noise $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. Furthermore, the n training inputs $\boldsymbol{x}_i \in \mathbb{R}^d$ were row-wise summarized in $\boldsymbol{X} \in \mathbb{R}^{n \times d}$. Our goal is the construction of a sparse GPR model which estimates the relationship above equivalent to [9].

Moreover, let I be the index set of size m of all active points $\boldsymbol{x}_i$ with $i \in I$, i.e. training points that represent the sparse approximation, and R be the index set of all remaining points, such that $I \cup R = \{1, \ldots, n\}$. The centered Gaussian prior density over the latent function values $\boldsymbol{f}_I \in \mathbb{R}^m$ corresponding to the active subset is then given by

$$p(\boldsymbol{f}_I \,|\, \boldsymbol{X}_I) = \mathcal{N}(\boldsymbol{f}_I \,|\, \boldsymbol{0},\ \boldsymbol{K}_I) \tag{1}$$

with $\boldsymbol{K}_I = \big(k(\boldsymbol{x}_i, \boldsymbol{x}_j)\big)_{i,j \in I} \in \mathbb{R}^{m \times m}$. Here, $\boldsymbol{K}_I$ is the covariance matrix over the active training points determined through the specified covariance function $k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The sparseness of this method is introduced via a likelihood approximation $q(\boldsymbol{y} \,|\, \boldsymbol{f}_I, \boldsymbol{X})$ that is optimized with respect to the Kullback-Leibler divergence (KL-divergence) and induced through the active training points which leads to

$$q_I(\boldsymbol{y} \,|\, \boldsymbol{f}_I, \boldsymbol{X}) = \mathcal{N}\big(\boldsymbol{y} \,\big|\, \boldsymbol{P}_I^T \boldsymbol{f}_I,\ \sigma^2 \boldsymbol{I}\big)\ . \tag{2}$$

The projection matrix $\boldsymbol{P}_I = \boldsymbol{K}_I^{-1} \boldsymbol{K}_{I,\cdot} \in \mathbb{R}^{m \times n}$, where $\boldsymbol{K}_{I,\cdot} \in \mathbb{R}^{m \times n}$ comprises the covariance function values between all training points ($\cdot$ notation) and the active subset of training points, maps $\boldsymbol{f}_I$ to the prior conditional mean $E\big[p_I(\boldsymbol{f} | \boldsymbol{f}_I)\big] = \boldsymbol{K}_{I,\cdot}^T \boldsymbol{K}_I^{-1} \boldsymbol{f}_I \in \mathbb{R}^n$. With Bayesian inference we get the approximated posterior density

$$q_I(\boldsymbol{f}_I \,|\, \boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}\big(\boldsymbol{f}_I \,\big|\, \boldsymbol{L}\boldsymbol{M}^{-1}\boldsymbol{V}\boldsymbol{y},\ \sigma^2 \boldsymbol{L}\boldsymbol{M}^{-1}\boldsymbol{L}^T\big) \tag{3}$$

which is proportional to the product of the prior (1) and the approximated likelihood (2). Here, $\boldsymbol{L} \in \mathbb{R}^{m \times m}$ is the lower Cholesky factor of $\boldsymbol{K}_I$, $\boldsymbol{V} = \boldsymbol{L}^{-1} \boldsymbol{K}_{I,\cdot} \in \mathbb{R}^{m \times n}$, and $\boldsymbol{M} = \sigma^2 \boldsymbol{I} + \boldsymbol{V}\boldsymbol{V}^T \in \mathbb{R}^{m \times m}$ for fixed I of size m. The approximated marginal likelihood directly follows from the integration over the same product about the active function values $\boldsymbol{f}_I$ and results in

$$q_I(\boldsymbol{y} \,|\, \boldsymbol{X}) = \mathcal{N}\big(\boldsymbol{y} \,\big|\, \boldsymbol{0},\ \sigma^2 \boldsymbol{I} + \boldsymbol{V}^T \boldsymbol{V}\big)\ . \tag{4}$$

The approximated posterior density for all training points $q_I(\boldsymbol{f} \,|\, \boldsymbol{y}, \boldsymbol{X})$ induced through the active subset is given by

$$q_I(\boldsymbol{f} \,|\, \boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}\big(\boldsymbol{f} \,\big|\, \boldsymbol{K}_{I,\cdot}^T \boldsymbol{a}_I,\ \boldsymbol{K} - \boldsymbol{V}^T\boldsymbol{V} + \sigma^2 \boldsymbol{V}^T \boldsymbol{M}^{-1} \boldsymbol{V}\big) \tag{5}$$

with mean vector $\boldsymbol{c}_I = E\big[q_I(\boldsymbol{f} | \boldsymbol{y}, \boldsymbol{X})\big] = \boldsymbol{K}_{I,\cdot}^T \boldsymbol{a}_I \in \mathbb{R}^n$. The predictive density for a test point $\boldsymbol{x}_*$ results in

$$\begin{aligned} q_I(\boldsymbol{f}_* \,|\, \boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}\Big(\boldsymbol{f}_* \,\Big|\, & \boldsymbol{k}_{I,*}^T \boldsymbol{L}^{-T} \boldsymbol{L}_M^{-T} \boldsymbol{b}_I,\ k_{**} - \big\|\boldsymbol{L}^{-1}\boldsymbol{k}_{I,*}\big\|^2 \\ & + \sigma^2 \big\|\boldsymbol{L}_M^{-1}\boldsymbol{L}^{-1}\boldsymbol{k}_{I,*}\big\|^2\Big) \end{aligned} \tag{6}$$

with the Cholesky decomposition $\boldsymbol{M} = \boldsymbol{L}_M \boldsymbol{L}_M^T$, $\boldsymbol{b}_I = \boldsymbol{L}_M^{-1} \boldsymbol{V}\boldsymbol{y} \in \mathbb{R}^m$, and the covariance vector $\boldsymbol{k}_{I,*} \in \mathbb{R}^m$ between the test input and the active points. If only the predicted mean values are of interest, the prediction vector $\boldsymbol{a}_I = \boldsymbol{L}^{-T} \boldsymbol{L}_M^{-T} \boldsymbol{b}_I$ can be precomputed, which allows fast computations of mean values with only $O(md)$ costs. Note that this cost depends on the calculation of the vector $\boldsymbol{k}_{I,*}$ and, thus, on the specified covariance function, which is typically proportional to the input dimension d. The predictive variance is feasible in $O(m^2)$ if $d < m$. The training complexity of this sparse GPR model is $O(nm^2)$ and, thus, much cheaper than the full GPR for $m << n$.

B. Efficient Active Set Selection Using Maximum Error Greedy Criterion

For the successive inclusion of training points into the active subset it is necessary to update the Cholesky factors $\boldsymbol{L}$, $\boldsymbol{L}_M$, the matrix $\boldsymbol{V}$, respectively $\boldsymbol{K}_{I,\cdot}$, the vector $\boldsymbol{b}_I$, and the mean $\boldsymbol{c}_I$ of the posterior distribution (5). This can be achieved with iteratively updating the above listed matrices as shown in [9]. The costs for the sequential insertion of one selected training point in the m-th iteration are $O(nm)$.

Most of the GP approximation techniques differ in the way, how the active set $\boldsymbol{X}_I$ is selected [7]. Usually, the

remaining point that has the maximum gain with respect to an insertion criterion Δ_i is selected. One of the best selection criteria is proposed by Smola and Bartlett [4]. They select the remaining point that maximizes the posterior likelihood

$$\begin{aligned} p(\boldsymbol{a}\,|\,\boldsymbol{y},\boldsymbol{X}) &\propto p(\,\boldsymbol{y}\,|\,\boldsymbol{a},\boldsymbol{X})\,p(\boldsymbol{a}\,|\,\boldsymbol{X}) \\ &\propto \mathcal{N}\big(\,\boldsymbol{y}\,\big|\,\boldsymbol{K}\boldsymbol{a}\,,\,\sigma^2\boldsymbol{I}\big)\,\mathcal{N}\big(\boldsymbol{a}\,\big|\,\boldsymbol{0}\,,\,\boldsymbol{K}^{-1}\big) \end{aligned} \tag{7}$$

for the admission of the prediction vector $\boldsymbol{a} \in \mathbb{R}^n$ of the full GP under the given data set $(\boldsymbol{y},\boldsymbol{X})$. This is based on the transformation of $\boldsymbol{a} = \boldsymbol{K}^{-1}\boldsymbol{f}$ and leads to the equivalent formulation

$$\tau_I = \min_{\boldsymbol{a}_I}\left(\frac{1}{2}\boldsymbol{a}_I^T\boldsymbol{L}\boldsymbol{M}\boldsymbol{L}^T\boldsymbol{a}_I - \boldsymbol{a}_I^T\boldsymbol{L}\boldsymbol{V}\boldsymbol{y}\right) = -\frac{1}{2}\boldsymbol{b}_I^T\boldsymbol{b}_I \tag{8}$$

in the sparse sense as pointed out in [9], i.e. $\boldsymbol{a}_R = \boldsymbol{0}$. In the following, let $I' = I \cup \{i\}$. The decrease in the sparse posterior likelihood derived from (7) defines the selection criterion by Smola and Bartlett (SB), i.e.

$$_{SB}\Delta_i = \tau_I - \tau_{I'} = \frac{1}{2}b_{I',i}^2 \;, \tag{9}$$

for a remaining point and with the new component $b_{I',i}$ of the updated vector $\boldsymbol{b}_{I'}$. The high computational costs of $O(nm)$ per criterion calculation for only one remaining training point are caused through a complete model update to become $\boldsymbol{b}_{I',i}$. Depending on the implementation, this model update can increase the memory requirements extremely for the remaining training point $\boldsymbol{x}_i$. Thus, the algorithm proposes evaluating the selection criterion only for a randomly chosen subset of remaining points with size r to keep the computational complexity feasible [4]. Nevertheless, the approach ends up with computational costs of $O(rnm^2)$ for the whole DTC approximation, which is the r-fold of the standard method with a randomized active set selection. Due to the randomized procedure involved when selecting r remaining points, the model performance in hard regression tasks can be suboptimal, especially for small r. Our proposed approach intends to further reduce the computational complexity of the selection procedure by making additional reasonable assumptions, while avoiding random selection as performed by Smola and Bartlett [4].

Similar to this method, our proposed approach maximizes also the posterior probability (7) with equation (8). The greedy scheme [4] successively maximizes the Euclidean norm of the vector $\boldsymbol{b}_{I'}$. This task is equivalent to iteratively minimizing $\left\|\boldsymbol{y}-\boldsymbol{c}_{I'}\right\|^2$ for the normalized vector $\boldsymbol{y}$ and, thus, approximately normalized $\boldsymbol{c}_{I'}$, since we have $\|\boldsymbol{b}_{I'}\|^2 = \boldsymbol{b}_{I'}^T\boldsymbol{b}_{I'} = \boldsymbol{y}^T\boldsymbol{c}_{I'}$ after a point inclusion. Due to the equivalence of norms in finite dimensional spaces, it holds true that $\left\|\boldsymbol{y}-\boldsymbol{c}_{I'}\right\| \leq n \max_{\forall j}|y_j - c_{I',j}|$. With the additional convergence assumption that in the limit, i.e. with increasing m we have $\boldsymbol{c}_{I'} \approx \boldsymbol{c}_I$, the costly update of the posterior model (5) for each remaining point can be alleviated. Thus, the employed assumption states that with increasing number of inserted points m, the resulting estimated mean $\boldsymbol{c}_{I'}$ does not significantly change. This assumption is intuitively meaningful and is empirically consistent with the convergence behavior during the learning process. Employing this assumption, the resulting new selection criterion can be given as

$$_{ME}\Delta_i = |y_i - c_{I,i}| \;. \tag{10}$$

Hence, our criterion selects the remaining points based on the maximal error (ME) under the current posterior model (5). This criterion is straightforward and computationally cheap to perform. It turns out, that our selection criterion leads to the $\mathcal{L}_1$-norm between the given observations and the mean of the current sparse GPR model. Our computationally efficient approach has $O(1)$ costs for criterion calculation per remaining point. This method results into a favorable hyperparameter learning process presented in the next section.

C. Incremental Learning of Hyperparameters

As the sparse GPR model depends on hyperparameters induced through the used covariance function, we present an incremental hyperparameter learning approach in this section, which can be employed while incrementally inserting data points to the active set. Let the vector $\boldsymbol{h}$ denote the collection of all hyperparameters. For notational simplicity the dependency of the above formulas on $\boldsymbol{h}$ was neglected. The adaptation of the hyperparameters can be realized by gradient-based optimization algorithms that minimize the negative log marginal likelihood obtained from (4), i.e.

$$\begin{aligned} -\log\big(q_I(\,\boldsymbol{y}\,|\,\boldsymbol{X},\boldsymbol{h})\big) = \;& \frac{n-m}{2}\log\big(\sigma^2\big) + \sum_{i=1}^{m}\log(l_{M,ii}) \\ & + \frac{1}{2\sigma^2}\left(\boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{b}_I^T\boldsymbol{b}_I\right) + \frac{n}{2}\log(2\pi)\;. \end{aligned} \tag{11}$$

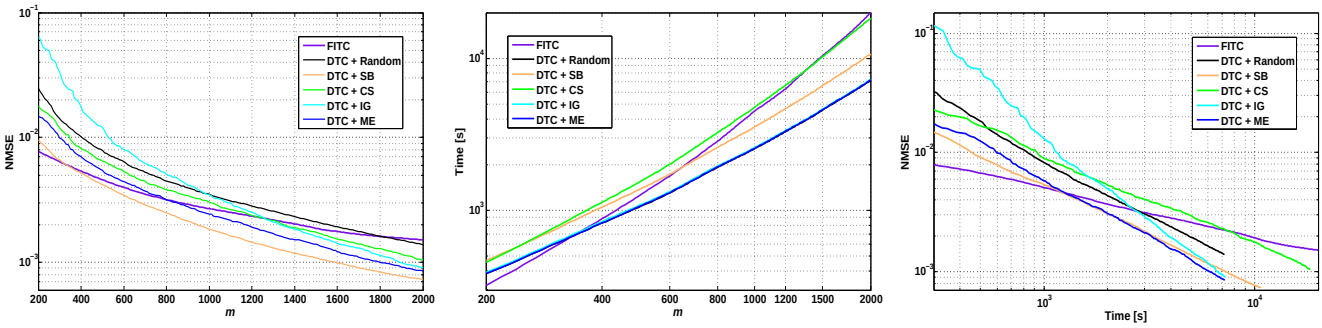
The values $l_{M,ii}$ are the diagonal entries of the Cholesky factor $\boldsymbol{L}_M$. One problem encountered when minimizing the negative log marginal likelihood (11) is their dependence on the active subset of training points. To solve this problem, we take alternating constrained optimization steps in an expectation maximization manner using the theory of [11]. Therefore, let $\mathcal{Z}$ be an i.i.d. Bernoulli random vector of size n which represents the hidden variables. The relationship between a realization $\boldsymbol{z}$ and I is that I contains the indices i of all z_i different from zero. From [11], it follows that the expectation step for estimating the new posterior density $q\left(\boldsymbol{z}^{t+1}\,|\,\boldsymbol{y},\boldsymbol{X},\boldsymbol{h}\right)$ is given over the minimization with respect to the KL-divergence by

$$\underset{q(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{h})\in q(\boldsymbol{y},\boldsymbol{X},\boldsymbol{h})}{\arg\min}\; KL\left[q(\boldsymbol{z}\,|\,\boldsymbol{y},\boldsymbol{X},\boldsymbol{h})\,\big\|\,q\left(\boldsymbol{y},\boldsymbol{z}\,|\,\boldsymbol{X},\boldsymbol{h}^t\right)\right]\;. \tag{12}$$

Here, the posterior distribution $q(\boldsymbol{z}\,|\,\boldsymbol{y},\boldsymbol{X},\boldsymbol{h})$ in the expectation step regarding to the KL-divergence is conditioned on the probability family $q(\boldsymbol{y},\boldsymbol{X},\boldsymbol{h})$ with $E\left[\sum_{i=1}^n z_i\right] = m$, i.e. the approximated posterior is induced by an active subset of size m. This condition is handled with a fixed final size m of the active subset in the greedy selection process. Furthermore, the maximization step results in

$$\underset{\boldsymbol{h}}{\arg\max}\; E_{\boldsymbol{z}^{t+1}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{h}}\left[\log\left(q(\boldsymbol{y},\boldsymbol{z}\,|\,\boldsymbol{X},\boldsymbol{h})\right)\right] \tag{13}$$

to determine an updated set of hyperparameters $\boldsymbol{h}^{t+1}$. In iteration step $t+1$ the maximization step is realized with



(a) NMSE of prediction errors for a different number of active points m.

(b) Complete learning time for a different number of active points m.

(c) Complete learning time vs. NMSE of prediction errors.

Fig. 2. Prediction errors as NMSE for the first DoF of the SARCOS arm is shown in 2(a), where the NMSE curves are usually a function of m. It can be seen that our approach *DTC+ME* is competitive to other DTC approximation techniques. For large number of m, *DTC+ME* converges to the best performing approach by Smola and Bartlett (*DTC+SB*). The learning time for the sparse GP approximations is shown in 2(b). Here, our *DTC+ME* is as fast as Seeger's information gain (*DTC+IG*) and stays close to the randomized insertion approach. Plot 2(c) shows that our method outperforms all other DTC type approaches in term of learning speed for sufficiently accurate models, i.e. small NMSE. Note that some of the axes are logarithmically scaled.

only few gradient ascents on the log marginal likelihood (11) on a fixed index set I determined by z. In this case, the repeated alternating computation of the E- and M-steps leads to a generalized EM algorithm, since we only increase the log marginal likelihood (11). Since the generalized EM algorithm converges to local maxima, the choice of the active training points is important in order to obtain a good set of hyperparameters $\boldsymbol{h}$. For the selection of the active subset, we employ our efficient maximum error criterion, as given in (10), to keep the hyperparameter learning fast and stable.

IV. Evaluations

In this section, we compare our maximum error (ME) selection criterion against other state-of-the-art methods for the DTC approximation. Especially, we compare our approach with methods mentioned in Section II, i.e. the information gain (IG) approach by Seeger et al. [9], the selection technique by Csató (CS) and Opper [10], the method by Smola and Bartlett (SB), and the purely randomized scheme. Furthermore, we also consider the FITC approximation, which is another class of GP approximation techniques, given by [8] for comparison. The comparison is performed on learning inverse dynamics using various robot data sets [5]. Subsequently, we use our proposed method to build an inverse dynamics model of a PR2 robot arm while employing it for a real-time, compliant tracking control task. For all experiments, we use the stationary squared exponential covariance function given through

$$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma_f^2 \exp\Big(-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{x}_j)^T \boldsymbol{\Lambda}^{-2}(\boldsymbol{x}_i-\boldsymbol{x}_j)\Big)\,, \qquad (14)$$

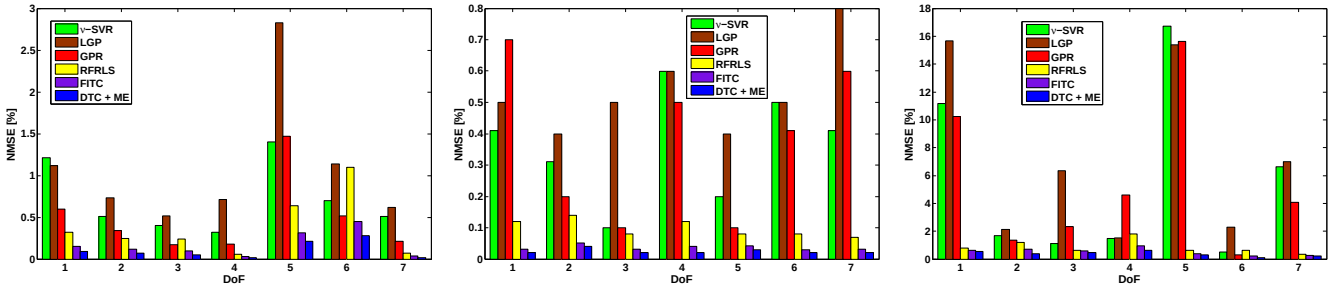
with signal magnitude σ_f^2 and the diagonal lenght-scales matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d\times d}$ as hyperparameters, see [1] for more details. The accuracy of the methods under consideration is measured by the normalized mean square error (NMSE). The NMSE is defined as the mean square error (MSE) divided by the variance of the targets $\boldsymbol{y}$ [5].

A. Comparisons on Inverse Dynamics Modeling

For validation, we employ a real data set from the SARCOS master arm (13922 training and 5569 test points), a simulation data set from the SARCOS model (14904 training and 5520 test points) and, finally, real robot data from the Barrett WAM arm (12000 training and 3000 test points), see [2], [5] for more details. Each point of the data sets has 21 input dimensions, i.e. position, velocity and acceleration of the 7 degrees of freedoms (DoF's), and 7 targets, i.e. one torque for each DoF of the SARCOS and Barrett robot arms.

In Fig. 2(a) we show the NMSE prediction errors as a function of the active set size for the DTC approximations, including our proposed approach and the FITC approximation on the first DoF using the real SARCOS data. We choose the first DoF of the SARCOS arm for this experiment, as it is quite hard to model, due to some latent non-linearities in the dynamics. For all DTC type approaches and the randomized active point selection, the size of active point sets is varied from 200 to 2000. To allow for a fair comparison, we adapt the number of gradient steps in the FITC approximation linearly with increasing virtual training points, i.e. we use $150+\frac{m}{4}$ optimization steps, because the number of hyperparameters in the FITC approximation also grows linearly with m. The NMSE results for random selection in the DTC approximation are averaged over ten runs.

As shown by the result, the maximum error approach (*DTC+ME*) is competitive to other DTC approximation techniques, while staying close to the best performing approach by Smola and Bartlett (*DTC+SB*). Especially, for sufficiently large number of active points, *DTC+ME* converges towards the performance of *DTC+SB*. In practice, the number of active points should be chosen as large as possible in order to ensure a good approximation. Considering the learning speed, *DTC+ME* is consistently faster than *DTC+SB*, while staying close the speed of randomize selection, as shown by Fig. 2(b). The NMSE results and the complete learning times in Fig. 2(b) are captured for the range between 200 to 2000 active points for all learning curves. Thus, the proposed approach presents a trade-off between learning



(a) NMSE on real SARCOS data (b) NMSE on SL simulated SARCOS data (c) NMSE on real Barrett WAM data

Fig. 3. NMSE diagrams (in percent) for each degree of freedom (DoF) after prediction on the test sets with real SARCOS data 3(a), SL simulation data 3(b), and robot data from the Barrett WAM arm 3(c). Overall, the DTC approximation with the maximum error (ME) criterion performs best, closely followed by the FITC approximation. The full GP does not perform well on these large data sets, due to suboptimal hyperparameters optimized from a subset of the training data [5].

speed and accuracy. This trade-off is additionally shown in Fig. 2(c), where the model accuracy is plotted against learning speed. For sufficiently accurate models, i.e. small NMSE values, *DTC+ME* requires less training time than other approximation techniques. Thus, our maximum error approach outperforms all DTC selection criteria with respect to training times for low NMSE values on test data, as shown in Fig. 2(c).

In Fig. 3, we compare our maximum error selection method for the sparse DTC approximation *DTC+ME* and the FITC approximation [8] with other established regression procedures [5] on all DoF's of real and simulated SARCOS test data, as well as real Barrett WAM data. The learning results for the other methods, i.e. local Gaussian processes (LGP), ν-SVR, GPR, and random Fourier regularized least squares (RFRLS), are taken from [5] and [12]. To allow for a fair comparison, we also employ a final set size of $m = 2000$ active and virtual training points for both *DTC+ME* and FITC approximations, respectively. We use 10 generalized EM steps for hyperparameter learning of the DTC approximation and 650 gradient ascents for the optimization of virtual training points with subsequent prediction.

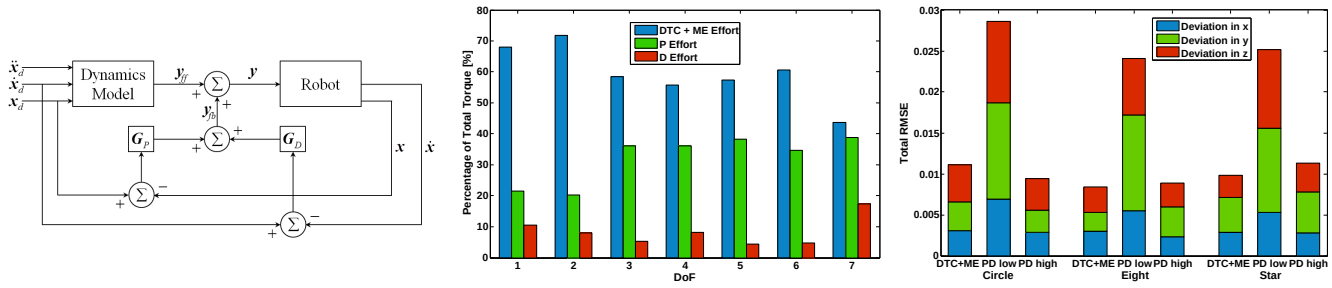
It can be observed that *DTC+ME* performs well on all provided data sets, and return better results than standard full GPR. The reason is that the hyperparameters of the standard GPR is optimized using a subset of the original data sets only, due to the costly optimization of the marginal likelihood [1]. This is a common procedure in order to reduce the computational cost for hyperparameter optimization when using large data for standard GP [1], [5]. However, depending on the subset selection for the hyperparameter optimization, the learned hyperparameters might be suboptimal and do not necessarily reflect the global data structure. Here, the incremental training of the hyperparameters during the selection process using EM, as shown in Section III-C, might present a good alternative. Compared to LGP, it should be noted that LGP is designed for applications in online real-time learning, where the learning speed is more important than accuracy and, thus, is not competitive in an offline comparison.

B. Compliant, Real-time Tracking Control

In this section, we employ learned inverse dynamics models for tracking control on a PR2 robot, as shown in Fig. 1. Here, the model-based tracking control law determines the joint torques $\boldsymbol{y}$ necessary for following a desired joint trajectory $\boldsymbol{x}_d$, $\dot{\boldsymbol{x}}_d$, $\ddot{\boldsymbol{x}}_d$, where $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\ddot{\boldsymbol{x}}$ are joint angles, velocities and accelerations of the robot, as shown in Fig. 4(a). This control paradigm uses a dynamics model, while employing feedback in order to stabilize the system. Here, the dynamics model of the robot can be used as a feed-forward model that predicts the joint torques $\boldsymbol{y}_{ff}$ required to perform the desired trajectory, while a feedback term $\boldsymbol{y}_{fb}$ ensures the stability of the tracking control with a resulting control law of $\boldsymbol{y} = \boldsymbol{y}_{ff} + \boldsymbol{y}_{fb}$. The feedback term can be a linear control law such as $\boldsymbol{y}_{fb} = \boldsymbol{G}_P \boldsymbol{e} + \boldsymbol{G}_D \dot{\boldsymbol{e}}$, where $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$ denotes the tracking error and $\boldsymbol{G}_P, \boldsymbol{G}_D$ position-gain and velocity-gain, respectively. If an accurate inverse dynamics model can be obtained, the feed-forward term $\boldsymbol{y}_{ff}$ will largely cancel the robots non-linearities [13]. In this case, $\boldsymbol{G}_P$ and $\boldsymbol{G}_D$ can be chosen to have small values enabling compliant control performance [2].

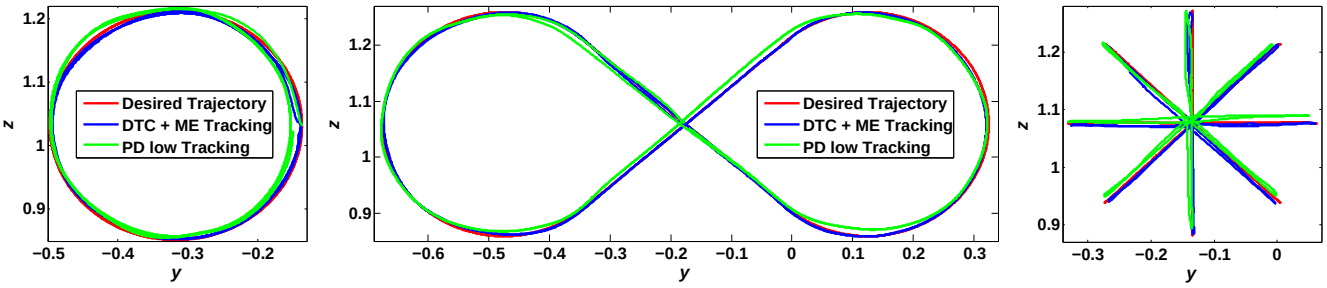
To obtain a global and precise dynamics model, we sample 517.783 data points with a frequency of $100Hz$ from the right arm of the PR2 robot. Furthermore, we train for each of the seven DoF's a sparse GP model given through the DTC approximation with our maximum error criterion. Thereby, the hyperparameters were always learned with 10 generalized EM steps, as explained in Section III-C. We choose a final active set size of $m = 1000$, which is sufficient to reach a good model quality while yielding prediction times below $3ms$ for all seven DoF's. Hereby, we are able to perform tracking control in real-time at $100Hz$.

In Fig. 4(b), we show the percentage on total torque for each DoF of the PR2 robot arm, where the gains for feedback control are chosen very small in order to enable compliant control. The contribution of the sparse, with *DTC+ME* approximated GP regression model to the control effort is usually far over 50%. A high contribution to the control effort indicates a good approximated model, as the feedback control loop does not need to strongly intervene during the control task here. The corresponding tracking performance in task-space is presented in Fig. 5 for three test trajectories, e.g. circle-, eight- and star-shape. Here, we compare the low gain feed-forward control using the learned dynamics model with the standard PD-control scheme. The results with respect to the RMSE value are shown in Fig. 4(c). It can be seen



(a) Tracking control using inverse dynamics model (b) Torque percentage for each DoF (c) Tracking error on the three test trajectories

Fig. 4. Plot 4(a) shows a scheme of feed-forward control using inverse dynamics model. Torque percentage for each DoF of the right PR2 arm is shown in Fig. 4(b). The higher the torque percentage, the more contributions have the corresponding parts. Here, *DTC+ME* dynamics models usually have far over 50% torque contribution. Plot 4(c) shows the tracking errors in task-space (x, y, z) for the three test trajectories, i.e. circle-, eight- and star-shape. The RMSE value of each dimension is computed for 3 different control schemes, i.e. low-gain *DTC+ME* model-based control, PD-control with low gains, and PD-control with high gains, i.e. about four times higher gains as in the low gain control schemes.



(a) Circle-shapes (b) Eight-shapes (c) Star-shape

Fig. 5. Tracking performance on the three test trajectories using low-gain *DTC+ME* model-based controller and PD-controller with low gains.

that using learned dynamics models we obtain compliant tracking control, and at the same time have tracking accuracy comparable to the one of high gain PD-control. A video showing the compliant, real-time robot control on the PR2 is attached as supplemental material.

V. Conclusion

In this work we have proposed a efficient and straightforward greedy insertion scheme for sparse GPR or, more precisely, for the DTC approximation. Our criterion is based on the maximum error between model and training data and we provided justification for this choice. It leads to a stable and efficient way for automatic sparse model selection. We also provide an incremental approach for learning the resulting hyperparameters using generalized expectation maximization. Many experiments show that the proposed approach is competitive in learning performance while being fast in computation. Further results on a real PR2 robot show that the proposed method can be employed for compliant, real-time robot control.

References

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[2] D. Nguyen-Tuong, J. Peters, and M. Seeger, "Computed Torque Control with Nonparametric Regression Models," in *Proceedings of the 2008 American Control Conference (ACC)*, 2008, pp. 212–217.

[3] B. Bischoff, D. Nguyen-Tuong, H. van Hoof, A. McHutchon, C. E. Rasmussen, A. Knoll, J. Peters, and M. P. Deisenroth, "Policy Search for Learning Robot Control Using Sparse Data," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[4] A. J. Smola and P. L. Bartlett, "Sparse Greedy Gaussian Process Regression," in *Advances in Neural Information Processing Systems (NIPS)*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., vol. 13, 2001, pp. 619–625.

[5] 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 (NIPS)*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21, 2009, pp. 1193–1200.

[6] C. K. I. Williams and M. Seeger, "Using the Nyström Method to Speed up Kernel Machines," in *Advances in Neural Information Processing Systems (NIPS)*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., vol. 13, 2001, pp. 682–688.

[7] J. Quiñonero-Candela and C. E. Rasmussen, "A Unifying View of Sparse Approximate Gaussian Process Regression," in *Journal of Machine Learning Research (JMLR)*, R. Herbrich, Ed., vol. 6, 2005, pp. 1939–1959.

[8] E. L. Snelson and Z. Ghahramani, "Sparse Gaussian Processes Using Pseudo-Inputs," in *Advances in Neural Information Processing Systems (NIPS)*, Y. Weiss, B. Schölkopf, and J. Platt, Eds., vol. 18, 2006, pp. 1257–1264.

[9] M. Seeger, C. K. I. Williams, and N. D. Lawrence, "Fast Forward Selection to Speed up Sparse Gaussian Process Regression," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, C. M. Bishop and B. J. Frey, Eds., 2003, pp. 205–212.

[10] L. Csató and M. Opper, "Sparse Representation for Gaussian Process Regression," in *Advances in Neural Information Processing Systems (NIPS)*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., vol. 13, 2001, pp. 444–450.

[11] J. V. Graça, K. Ganchev, and B. Taskar, "Expectation Maximization and Posterior Constraints," in *Advances in Neural Information Processing Systems (NIPS)*, J. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds., vol. 20, 2008, pp. 569–576.

[12] A. Gijsberts and G. Metta, "Incremental Learning of Robot Dynamics Using Random Features," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 951–956.

[13] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control*. John Wiley & Sons, 2004, vol. 2.