Prediction under Uncertainty in Sparse Spectrum Gaussian Processes with Applications to Filtering and Control

Yunpeng Pan
School of Aerospace Engineering
Georgia Institute of Technology
ypan37@gatech.edu

Xinyan Yan
School of Interactive Computing
Georgia Institute of Technology
xyan43@gatech.edu

Evangelos A. Theodorou
School of Aerospace Engineering
Georgia Institute of Technology
evangelos.theodorou@gatech.edu

Byron Boots
School of Interactive Computing
Georgia Institute of Technology
bboots@cc.gatech.edu

Abstract

In many sequential prediction and decision-making problems such as Bayesian filtering and probabilistic model-based planning and control, we need to cope with the challenge of *prediction under uncertainty*, where the goal is to compute the predictive distribution $p(y)$ given a input distribution $p(x)$ and a probabilistic model $p(y|x)$. Computing the exact predictive distribution is generally intractable. In this work, we consider a special class of problems in which the input distribution $p(x)$ is a multivariate Gaussian, and the probabilistic model $p(y|x)$ is learned from data and specified by a sparse spectral representation of Gaussian processes (SSGPs).

SSGPs are a powerful tool for scaling Gaussian processes (GPs) to large datasets by approximating the covariance function using finite-dimensional random Fourier features. Existing SSGP algorithms for regression assume deterministic inputs, precluding their use in many sequential prediction and decision-making applications where accounting for input uncertainty is crucial. To address this *prediction under uncertainty* problem, we propose an exact moment-matching approach with closed-form expressions for predictive distributions. Our method is more general and scalable than its standard GP counterpart, and is naturally applicable to multi-step prediction or uncertainty propagation. We show that our method can be used to develop new algorithms for Bayesian filtering and stochastic model predictive control, and we evaluate the applicability of our method with both simulated and real-world experiments.

Keywords: Sequential prediction, Gaussian processes, planning and control, Bayesian filtering, probabilistic model-based reinforcement learning, approximate inference

1 Introduction

The problem of *prediction under uncertainty*, appears in many fields of science and engineering that involve sequential prediction and decision-making including state estimation, time series prediction, stochastic process approximation, and planning and control. In these problems, uncertainty can be found in both the predictive models and the model's inputs. Formally, we are often interested in finding the probability density of a prediction y, given a distribution $p(x)$ and a probabilistic model $p(y|x)$. By marginalization,

$$p(y) = \int p(y|x)p(x)\,\mathrm{d}x. \tag{1}$$

Unfortunately, computing this integral exactly is often intractable. In this paper, we tackle a subfamily of (1) where: 1) the probabilistic model is learned from data and specified by a sparse spectrum representation of a Gaussian process (SSGP); and 2) the input x is normally distributed.

1.1 Related work

Gaussian Process (GP) inference with uncertain inputs has been addressed by [2, 8], and extended to the multivariate outputs by [10]. These methods have led to the development of many algorithms in reinforcement learning [15, 4], Bayesian filtering [9, 6], and smoothing [5]. However, these approaches have two major limitations: 1) they are not directly applicable to large datasets, due to the polynomial (in data samples) time complexity for exact inference [16]; and 2) analytic moment expressions, when used, are restricted to squared exponential (SE) or polynomial kernels [10] and cannot be generalized to other kernels in a straightforward way.

A common method for approximating large-scale kernel machines is through random Fourier features [14]. The key idea is to map the input to a low-dimensional feature space yielding fast linear methods. In the context of GP regression (GPR), this idea leads to the sparse spectrum GPR (SSGPR) algorithm [11]. SSGP has been extended in a number of ways for, e.g. incremental model learning [7], and large-scale GPR [3]. However, to the best of our knowledge, prediction under uncertainty for SSGPs has not been explored. Although there are several alternative approximations to exact GP inference including approximating the posterior distribution using inducing points, comparing different GP approximations is not the focus of this paper.

1.2 Applications

We consider two problems that involve sequential prediction and decision-making: Bayesian filtering and stochastic model predictive control (MPC). The goal of Bayesian filtering is to infer a hidden system state from observations through the recursive application of Bayes' rule. GP-based assumed density filtering (ADF) with SE kernels has been developed by [6], which has demonstrated superior performance compared to other GP-based filters [9]. We extend this work with a highly efficient SSGP-based ADF approach.

The goal of stochastic MPC is to find finite horizon optimal control at each time instant. Due to the high computational cost of GP inference and real-time optimization requirements in MPC, most GP-based control methods [4, 13] are restricted to episodic reinforcement learning tasks. To cope with this challenge, we present an SSGP-based MPC approach that is fast enough to perform optimization and model adaptation on-the-fly.

1.3 Our contributions

- We propose an exact moment-matching approach to prediction under uncertainty in SSGPs with closed-form expressions for the predictive distribution. Compared to previous GP counterparts, our method: 1) is more scalable, and 2) can be generalized to any continuous shift-invariant kernels with a Fourier feature representation.
- We demonstrate successful applications of the proposed approach to 1) recursive Bayesian filtering and 2) stochastic model predictive control.

2 Sparse Spectral Representation of Gaussian Processes

Consider the task of learning the function $f : \mathbf{R}^d \to \mathbf{R}$, given data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, with each pair related by

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2), \tag{2}$$

where ϵ is IID additive Gaussian noise. Gaussian process regression (GPR) is a principled way of performing Bayesian inference in function space, assuming that function f has a prior distribution $f \sim \mathcal{GP}(m, k)$, with mean function $m : \mathbf{R}^d \to \mathbf{R}$ and kernel $k : \mathbf{R}^d \times \mathbf{R}^d \to \mathbf{R}$. Without loss of generality, we assume $m(x) = 0$. Exact GPR is challenging for large datasets due to its $O(n^3)$ time and $O(n^2)$ space complexity [16], which is a direct consequence of having to store and invert an $n \times n$ Gram matrix.

Random features can be used to form an unbiased approximation of continuous shift-invariant kernel functions, and are proposed as a general mechanism to accelerate large-scale kernel machines [14], via explicitly mapping inputs to low-dimensional feature space. Based on Bochner's theorem, the Fourier transform of a continuous shift-invariant positive definite kernel $k(x, x')$ is a proper probability distribution $p(\omega)$ [14], which leads to an unbiased approximation of k: $k(x, x') \approx \frac{1}{m} \sum \phi_{\omega_i}(x)\phi_{\omega_i}(x')^*$, where random frequencies $\{\omega_i\}_{i=1}^m$ are drawn IID from $p(\omega)$. Considering ϕ_ω can be replaced by sinusoidal functions since both $p(\omega)$ and $k(x, x')$ are reals, and concatenating features $\{\phi_{\omega_i}\}_{i=1}^m$ into a succinct vector form, an approximation for $k(x, x')$ is expressed as:

$$k(x,x') \approx \phi(x)^T\phi(x'),\ \phi(x) = \begin{bmatrix}\phi^c(x)\\ \phi^s(x)\end{bmatrix}, \phi_i^c(x) = \sigma_k \cos(\omega_i^T x),\ \phi_i^s(x) = \sigma_k \sin(\omega_i^T x),\ \omega_i \sim p(\omega), \tag{3}$$

where σ_k is a scaling coefficient. For the commonly used Squared Exponential (SE) kernel: $k(x,x') = \sigma_f^2 \exp(-\frac{1}{2}\|x-x'\|_{\Lambda^{-1}})$, $p(\omega) = \mathcal{N}(0, \Lambda^{-1})$ and $\sigma_k = \frac{\sigma_f}{\sqrt{m}}$, where the coefficient σ_f and the diagonal matrix Λ are the hyperparameters. Spectral densities $p(\omega)$ and scaling coefficients σ_k for other continuous shift-invariant kernels can be derived similarly. Based on the feature map, SSGP is defined as

Definition 1. *Sparse Spectrum GPs (SSGPs) are GPs with kernels defined on the finite-dimensional and randomized feature map ϕ (3):*

$$\hat{k}(x,x') = \phi(x)^T\phi(x') + \sigma_n^2\delta(x-x'), \tag{4}$$

where the function δ is the Kronecker delta function, to account for the additive zero mean Gaussian noise in (2)*.*

Because of the explicit finite-dimensional feature map (3), each SSGP is equivalent to a Gaussian distribution over the weights of features $w \in \mathbf{R}^{2m}$. Assuming that prior distribution of weights w is $\mathcal{N}(0, \Sigma_w)$, and the feature map is fixed, after conditioning on the data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ the posterior distribution of w is

$$w \sim \mathcal{N}(\alpha,\ \sigma_n^2 A^{-1}), \quad \alpha = A^{-1}\Phi Y, \quad A = \Phi\Phi^T + \sigma_n^2\Sigma_\alpha^{-1}, \tag{5}$$

which can be derived through Bayesian linear regression. In (5), the column vector Y and the matrix Φ are specified by the data $\mathcal{D}$: $Y = [y_1 \quad \dots \quad y_n]^T$, $\Phi = [\phi(x_1) \quad \dots \quad \phi(x_n)]$. Consequently, the posterior distribution over the output y in (2) at a test point x is *exactly Gaussian*, in which the posterior variance explicitly captures the model uncertainty in prediction with input x:

$$p(y|x) = \mathcal{N}(\alpha^T\phi(x),\ \sigma_n^2 + \sigma_n^2\|\phi(x)\|^2_{A^{-1}}). \tag{6}$$

This SSGP regression method is proposed in [11]. Its time complexity is $O(nm^2 + m^3)$, which is significantly more efficient than standard GPR's $O(n^3)$ when $m \ll n$.

Remark: It's worth noting that the method proposed in this paper is not tied to specific algorithms for SSGP regression such as linear Bayesian regression [11], but accounts for any SSGP with specified feature weights distribution (5), where posterior α and A can be computed by any means. Variations on A include sparse approximations by a low rank plus diagonal matrix, or iterative solutions by optimization methods like doubly stochastic gradient descent [3].

3 Prediction under Uncertainty

We present an analytic moment-based approach to (1) under two conditions: 1) the uncertain input is normally distributed: $x \sim \mathcal{N}(\mu, \Sigma)$, and 2) probabilistic models are in the form of (6) specified by SSGPs. Despite these conditions, evaluating the integral in (1) is still intractable. In this work, we approximate the true predictive distribution $p(y)$ by a Gaussian via computation of the exact moments in closed-form. We consider multivariate outputs by utilizing conditionally independent scalar models for each output dimension, *i.e.*, assuming for outputs in different dimension y_a and y_b, $p(y_a, y_b|x) = p(y_a|x)p(y_b|x)$. For notational simplicity, we suppress the dependency of $\phi(x)$ on x, and treat y as a scalar by default.

In the following we present our method, SSGP-exact moment matching (SSGP-EMM). We derive 1) the predictive mean $\mathbf{E}\, y$; 2) the predictive variance $\mathbf{Var}\, y$ and covariance $\mathbf{Cov}(y_a, y_b)$, which in the multivariate case correspond to the diagonal and off-diagonal entries of the predictive covariance matrix; and 3) the cross-covariance between input and prediction $\mathbf{Cov}(x, y)$.

Using the expressions for SSGP (3), (6), and the law of total expectation, the predictive mean becomes

$$\mathbf{E}\, y = \mathbf{E}\,\mathbf{E}(y|x) = \mathbf{E}\left(\alpha^T\phi\right) = \alpha^T\,\mathbf{E}\begin{bmatrix}\phi^c\\ \phi^s\end{bmatrix}, \quad \mathbf{E}\,\phi_i^c = \sigma_k\,\mathbf{E}\cos(\omega_i^T x), \quad \mathbf{E}\,\phi_i^s = \sigma_k\,\mathbf{E}\sin(\omega_i^T x), \tag{7}$$

where $i = 1, \dots, m$, and in the nested expectation $\mathbf{E}\,\mathbf{E}(y|x)$, the outer expectation is over the input distribution $p(x) = \mathcal{N}(\mu, \Sigma)$, and the inner expectation is over the conditional distribution $p(y|x)$ (6).

By observing (7), we see that the expectation of sinusoids under the Gaussian distribution is the key to computing the predictive mean. Thus we state the following proposition:

Proposition 1. *The expectation of sinusoids over multivariate Gaussian distributions: $x \sim \mathcal{N}(\mu, \Sigma)$, $x \in \mathbf{R}^d$, i.e., $p(x) = (2\pi)^{-\frac{d}{2}}(\det\Sigma)^{-\frac{1}{2}}\exp(-\frac{1}{2}\|x-\mu\|^2_{\Sigma^{-1}})$, can be computed analytically:*

$$\mathbf{E}\cos(\omega^T x) = \exp(-\frac{1}{2}\|\omega\|^2_\Sigma)\cos(\omega^T\mu), \quad \mathbf{E}\sin(\omega^T x) = \exp(-\frac{1}{2}\|\omega\|^2_\Sigma)\sin(\omega^T\mu).$$

The proof is omitted. The predictive mean (7), variance, and covariance between different outputs are derived using Proposition 1. By the law of total variance, the predictive variance is

$$\mathbf{Var}\, y = \mathbf{E}\,\mathbf{Var}(y|x) + \mathbf{Var}\,\mathbf{E}(y|x)\alpha^T\Psi\alpha - (\mathbf{E}\, y)^2, \tag{8}$$

where Ψ is defined as the expectation of the outer product of feature vectors over input distribution $p(x)$. Specifically, we compute Ψ by applying the product-to-sum trigonometric identities:

$$\mathbf{E}\left(\phi\phi^T\right) = \Psi = \begin{bmatrix} \Psi^{cc} & \Psi^{cs} \\ \Psi^{sc} & \Psi^{ss} \end{bmatrix}, \qquad \Psi^{cc}_{ij} = \frac{\sigma_k^2}{2}\left(\mathbf{E}\left(\cos(\omega_i+\omega_j)^T x\right) + \mathbf{E}\left(\cos(\omega_i-\omega_j)^T x\right)\right),$$

$$\Psi^{ss}_{ij} = \frac{\sigma_k^2}{2}\left(\mathbf{E}\left(\cos(\omega_i-\omega_j)^T x\right) - \mathbf{E}\left(\cos(\omega_i+\omega_j)^T x\right)\right), \quad \Psi^{cs}_{ij} = \frac{\sigma_k^2}{2}\left(\mathbf{E}\left(\sin(\omega_i+\omega_j)^T x\right) - \mathbf{E}\left(\sin(\omega_i-\omega_j)^T x\right)\right) \tag{9}$$

where $\Psi^{cc}, \Psi^{ss}, \Psi^{cs}$ are $m \times m$ matrices, and $i, j = 1, \ldots, m$, on whose terms Proposition 1 can be directly applied.

Next we derive the covariance for different output dimensions for multivariate prediction. These correspond to the off-diagonal entries of the predictive covariance matrix. We show that, despite the conditional independence assumption for different outputs given a deterministic input, outputs become coupled with uncertain input. Using the law of total covariance, the covariance is:

$$\mathbf{Cov}(y_a, y_b) = \mathbf{Cov}\left(\mathbf{E}(y_a|x), \mathbf{E}(y_b|x)\right) = \mathbf{E}\left(\mathbf{E}(y_a|x), \mathbf{E}(y_b|x)\right) - (\mathbf{E}\, y_a)(\mathbf{E}\, y_b) = \alpha_a^T \Psi_{ab} \alpha_b - (\alpha_a^T\, \mathbf{E}\, \phi_a)(\alpha_b^T\, \mathbf{E}\, \phi_b) \tag{10}$$

where matrix Ψ_{ab} is the expectation of the outer product of feature vectors corresponding to different feature maps ϕ_a, ϕ_b for outputs y_a, y_b, computed similarly as in (3) with corresponding random frequencies $\{\omega_i\}$, and the scaling coefficient σ_k (3). α_a and α_b are the corresponding weight vectors for y_a and y_b (6). Compared to the expression for the variance of a single output in (8), the term $\mathbf{E}\left(\mathbf{Cov}(y_a|x)\, \mathbf{Cov}(y_b|x)\right)$ that is included in the law of total covariance is neglected due to the assumption of conditional independence of different outputs (§2), so (10) does not have the corresponding first two terms in (8).

Finally, we compute the covariance between input and each output dimension. Invoking the law of total covariance:

$$\mathbf{Cov}(x, y) = \mathbf{Cov}(x, \mathbf{E}(y|x)) = \mathbf{E}\left(x\, \mathbf{E}(y|x)\right) - (\mathbf{E}\, x)(\mathbf{E}\, y) = \Upsilon\alpha - (\mathbf{E}\, y)\mu, \tag{11}$$

where matrix Υ is the expectation of the outer product of the input x and the feature vector $\phi(x)$ over input distribution $x \sim \mathcal{N}(\mu, \Sigma)$:

$$\mathbf{E}(x\phi^T) = \Upsilon = [\Upsilon_1^c \quad \ldots \quad \Upsilon_m^c \quad \Upsilon_1^s \quad \ldots \quad \Upsilon_m^s], \quad \Upsilon_i^c = \sigma_k\, \mathbf{E}\left(\cos(\omega_i^T x)x\right), \quad \Upsilon_i^s = \sigma_k\, \mathbf{E}\left(\cos(\omega_i^T x)x\right),$$

where $i = 1, \ldots, m$. We state the following proposition to compute each column in Υ consisting of expectations of sinusoidal functions and inputs.

Proposition 2. *The expectation of the multiplication of sinusoids and linear functions over multivariate Gaussian distributions: $x \sim \mathcal{N}(\mu, \Sigma)$, can be computed analytically:*

$$\mathbf{E}\left(\cos(\omega^T x)x\right) = \left(\mathbf{E}\cos(\omega^T x)\right)\mu - (\mathbf{E}(\sin(\omega^T x))\Sigma\omega, \qquad \mathbf{E}\left(\sin(\omega^T x)x\right) = \left(\mathbf{E}\sin(\omega^T x)\right)\mu + \left(\mathbf{E}\cos(\omega^T x)\right)\Sigma\omega,$$

where the right-hand-side expectations have analytical expressions (Proposition 1).

The proof is omitted. Next, the result is extended to $\mathbf{E}\left(x\cos(\omega^T x)\right)$, by setting a to consist of indicator vectors. Applying Proposition 1 and 2, we complete the derivation of $\mathbf{Cov}(x, y)$ in (11).

Remark: SSGP-EMM's computation complexity is $O\left(m^2k^2d^2\right)$, where m is the number of features, k is the output dimension, and d is the input dimension. Compared to the multivariate moment-matching approach for GPs (GP-EMM) [8, 10] with $O\left(n^2k^2d^2\right)$ time complexity, SSGP-EMM is more efficient when $m \ll n$. Moreover, our approach is applicable to any positive-definite continuous shift-invariant kernel with different spectral densities, while previous approaches like GP-EMM [10] are only derived for squared exponential (SE) kernels [1].

Method	SSGP-EMM	GP-EMM
Time	$O(m^2k^2d^2)$	$O(n^2k^2d^2)$
Applicable kernels	continuous shift-invariant kernels	SE kernels

Table 1: Comparison of our proposed methods and GP-EMM [8, 10] in computational complexity and generalizability.

4 Applications to Bayesian Filtering and Predictive Control

We focus on the application of the proposed methods to Bayesian filtering and predictive control. We consider the following discrete-time nonlinear dynamical system:

$$x_{t+1} = f(x_t, u_t) + w_t, \;\; w_t \sim \mathcal{N}(0, \Sigma_w), \qquad y_t = g(x_t) + v_t, \;\; v_t \sim \mathcal{N}(0, \Sigma_v), \tag{12}$$

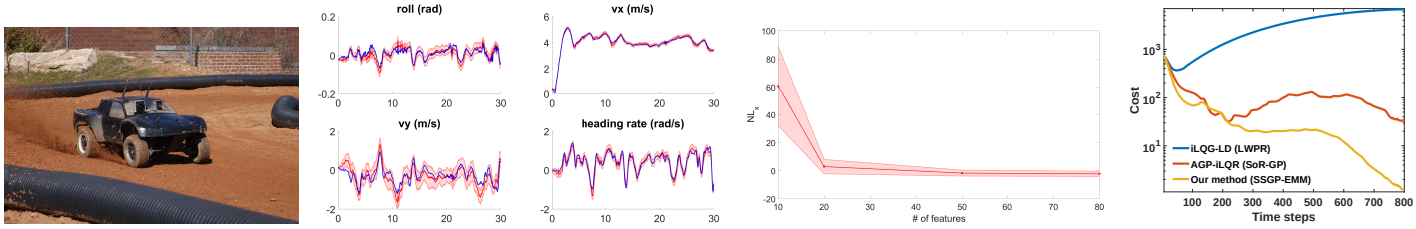
where $x_t \in \mathbf{R}^d$ is state, $u_t \in \mathbf{R}^r$ is control, $y_t \in \mathbf{R}^k$ is observation or measurement, $w_t \in \mathbf{R}^d$ is IID process noise, $v_t \in \mathbf{R}^k$ is IID measurement noise, and subscript t denotes discrete time index. We consider scenarios where f and g are unknown but a dataset $\mathcal{D} = \left(\{(x_t, u_t), x_{t+1}\}_{t=1}^{n-1}, \{x_t, y_t\}_{t=1}^{n}\right)$ is provided. The dynamics model $p(x_{t+1}|x_t, u_t)$ is learned using state transition pairs $\{(x_t, u_t), x_{t+1}\}_{t=1}^{n-1}$, and the observation model $p(y_t|x_t)$ is learned separately from state-observation pairs $\{x_t, y_t\}_{t=1}^{n}$.

Bayesian filtering: The task of Bayesian filtering is to infer the posterior distribution of the current state of a dynamical system based on the current and past noisy observations, *i.e.*, finding $p(x_{t|t})$, where the notation $x_{t|s}$ denotes the random variable $x_t|y_0, \ldots, y_s$. Due to the Markov property of the process x, *i.e.*, $x_t|x_0, \ldots, x_{t-1} = x_t|x_{t-1}$, in Gauss-Markov models, $p(x_{t|t})$ can be computed

[1] Expressions for polynomial kernels can be derived similarly.

recursively through an alternating *prediction step* and *correction step*. We use the proposed SSGP-EMM to propagate full densities thorough the probabilistic dynamics and observation models instead of using linearization or finite-samples [9]. Our method is related to GP-ADF [6] which is based on GP-EMM [8, 10]. However, our method is more scalable and general as discussed in §3. In order to demonstrate the applicability of our method, we use a real-world state estimation task in high-speed autonomous driving on a dirt track (Figure 1a). The goal is estimating the vehicle's linear velocities, heading rate, and roll angle using only wheel speed sensors and ground truth samples generated by integrating GPS and IMU data. Filtered distributions using 80 features are shown in Figure 1b, and Figure 1c shows the negative log-likelihood of state NL_x for different number of features. Surprisingly, only a small number of features is necessary for satisfactory results.

Model Predictive Control: The goal of stochastic model predictive control (MPC) is to choose a control sequence that minimizes the expected cost, given a dynamics model and cost function. The main challenge of applying MPC in practice is efficient and accurate multi-step prediction due to the lack of explicit and accurate models. GPs have been used for dynamics modeling and prediction in control algorithms to cope with model uncertainty [4, 13]. However, these methods are restricted to off-line optimization due to the computational burden of GP inference. On the other hand, more efficient methods usually drop the uncertainty in the probabilstic model or input (1), e.g., iLQG-LD [12] uses Locally Weighted Projection Regression (LWPR), AGP-iLQR uses subset of regressors (SoR) approximation for GPs [1]. In this work, we combine our proposed SSGP-EMM and trajectory optimization (similar to [1, 12, 13]) for MPC. In addition, the SSGP dynamics model is updated incrementally. We demonstrate the performance of our algorithm using a simulated robotic arm manipulation task, in which our goal is tracking a moving target under dynamics variations. We use 1200 samples and 50 features for model learning and compare our method with iLQG-LD [12] and AGP-iLQR [1]. Results are shown in Figure (1d). Our method outperforms the other two because our multi-step predictions are more robust to model error. More precisely, the other methods do not propagate the full densities through probabilistic models of the dynamics, as we do here.



(a) Autonomous driving task (b) Filtered distribution vs. ground truth (c) NL_x vs. # of features (d) Predictive control cost

Figure 1: (a) High-speed autonomous driving task. (b) All state trajectories, blue lines are the ground truth (30 seconds continuous driving), red areas are filtered distributions. In (c), red is the mean and variance of the negative log-likelihood NL_x for the 1200 filtering steps (30 seconds driving). (d) The total trajectory cost comparison for iLQG-LD [12], AGP-iLQR [1] and our method.

5 Conclusion

We introduced an analytic moment-based approach to *prediction under uncertainty* in sparse spectrum Gaussian processes (SSGPs). Compared to its full GP counterpart [8, 10], our method is more general in terms of the choice of kernels, and is more scalable thanks to the sparse spectrum representation (see Table 1). Although we adopt the name SSGP, our proposed method is not tied to specific model learning methods such as linear Bayesian regression [11]. Our method is directly applicable to many sequential prediction and decision-making problems that involve uncertain dynamical systems. We demonstrated the performance of our method in Bayesian filtering and predictive control tasks using both real-world and simulated experiments.

References

[1] J. Boedecker, JT. Springenberg, J. Wulfing, and M. Riedmiller. Approximate real-time optimal control based on sparse Gaussian process models. In *ADPRL 2014*.

[2] J. Quinonero Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of uncertainty in Bayesian kernel models-application to multiple-step ahead forecasting. 2003.

[3] Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina F Balcan, and Le Song. Scalable kernel methods via doubly stochastic gradients. In *NIPS*, 2014.

[4] M. Deisenroth, D. Fox, and C. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE PAMI*, 27:75–90, 2015.

[5] M. Deisenroth, R. Turner, MF Huber, UD Hanebeck, and CE Rasmussen. Robust filtering and smoothing with gaussian processes. *IEEE Trans. on Automatic Control*, 2012.

[6] Marc Peter Deisenroth, Marco F Huber, and Uwe D Hanebeck. Analytic moment-based gaussian process filtering. In *ICML*, 2009.

[7] A. Gijsberts and G. Metta. Real-time model learning using incremental sparse spectrum Gaussian process regression. *Neural Networks*, 41:59–69, 2013.

[8] A. Girard, C.E. Rasmussen, J. Quinonero-Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In *NIPS*, 2003.

[9] J. Ko and D. Fox. Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models. *Autonomous Robots*, 27(1):75–90, 2009.

[10] Malte Kuss. *Gaussian process models for robust regression, classification, and reinforcement learning*. PhD thesis, Technische Universität, 2006.

[11] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum gaussian process regression. *JMLR*, 99:1865–1881, 2010.

[12] D. Mitrovic, S. Klanke, and S. Vijayakumar. Adaptive optimal feedback control with learned internal dynamics models. 2010.

[13] Y. Pan and E. Theodorou. Probabilistic differential dynamic programming. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1907–1915, 2014.

[14] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

[15] C. Rasmussen and M. Kuss. Gaussian processes in reinforcement learning. In *NIPS*, volume 4, page 1, 2004.

[16] C.K.I Williams and C.E. Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.