Robust Filtering and Smoothing with Gaussian Processes

Marc Peter Deisenroth, Ryan Turner *Member, IEEE*,
Marco F. Huber *Member, IEEE*,
Uwe D. Hanebeck *Member, IEEE*, Carl Edward Rasmussen

***Abstract*—We propose a principled algorithm for robust Bayesian filtering and smoothing in nonlinear stochastic dynamic systems when both the transition function and the measurement function are described by non-parametric Gaussian process (GP) models. GPs are gaining increasing importance in signal processing, machine learning, robotics, and control for representing unknown system functions by posterior probability distributions. This modern way of "system identification" is more robust than finding point estimates of a parametric function representation. Our principled filtering/smoothing approach for GP dynamic systems is based on analytic moment matching in the context of the forward-backward algorithm. Our numerical evaluations demonstrate the robustness of the proposed approach in situations where other state-of-the-art Gaussian filters and smoothers can fail.**

***Index Terms*—Nonlinear systems, Bayesian inference, smoothing, filtering, Gaussian processes, machine learning**

I. Introduction

Filtering and smoothing in the context of dynamic systems refers to a Bayesian methodology for computing posterior distributions of the latent state based on a history of noisy measurements. This kind of methodology can be found, e.g., in navigation, control engineering, robotics, and machine learning [1]–[4]. Solutions to filtering [1]–[5] and smoothing [6]–[9] in linear dynamic systems are well known, and numerous approximations for nonlinear systems have been proposed, for both filtering [10]–[15] and smoothing [16]–[19].

In this note, we focus on Gaussian filtering and smoothing in Gaussian process (GP) dynamic systems. GPs are a robust non-parametric method for approximating unknown functions by a posterior distribution over them [20], [21]. Although GPs have been around for decades, they only recently became computationally interesting for applications in robotics, control, and machine learning [22]–[26].

The contribution of this note is the derivation of a novel, principled, and robust Rauch-Tung-Striebel (RTS) smoother for GP dynamic systems, which we call the *GP-RTSS*. The GP-RTSS computes a Gaussian approximation to the smoothing distribution in closed form. The posterior filtering and smoothing distributions can be computed *without* linearization [10] or sampling approximations of densities [11].

We provide numerical evidence that the GP-RTSS is more robust than state-of-the-art nonlinear Gaussian filtering and smoothing algorithms including the extended Kalman filter (EKF) [10], the unscented Kalman filter (UKF) [11], the cubature Kalman filter (CKF) [15], the GP-UKF [12], and their corresponding RTS smoothers. *Robustness* refers to the ability of an inferred distribution to explain the "true" state/measurement.

The paper is structured as follows: In Sections I-A, I-B, we introduce the problem setup and necessary background on Gaussian smoothing and GP dynamic systems. In Section II, we briefly introduce Gaussian process regression, discuss the expressiveness of a GP, and explain how to train GPs. Section III details our proposed method (GP-RTSS) for smoothing in GP dynamic systems. In Section IV, we provide experimental evidence of the robustness of the GP-RTSS. Section V concludes the paper with a discussion.

A. Problem Formulation and Notation

In this note, we consider discrete-time stochastic systems

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t \tag{1}$$

$$\mathbf{z}_t = g(\mathbf{x}_t) + \mathbf{v}_t \tag{2}$$

where $\mathbf{x}_t \in \mathbb{R}^D$ is the state, $\mathbf{z}_t \in \mathbb{R}^E$ is the measurement at time step t, $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_w)$ is Gaussian system noise, $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_v)$ is Gaussian measurement noise, f is the transition function (or system function) and g is the measurement function. The discrete time steps t run from 0 to T. The initial state $\mathbf{x}_0$ of the time series is distributed according to a Gaussian prior distribution $p(\mathbf{x}_0) = \mathcal{N}(\boldsymbol{\mu}_0^x, \boldsymbol{\Sigma}_0^x)$. The purpose of filtering and smoothing is to find approximations to the posterior distributions $p(\mathbf{x}_t|\mathbf{z}_{1:\tau})$, where $1\!:\!\tau$ in a subindex abbreviates $1, \dots, \tau$ with $\tau = t$ during filtering and $\tau = T$ during smoothing. In this note, we consider Gaussian approximations $p(\mathbf{x}_t|\mathbf{z}_{1:\tau}) \approx \mathcal{N}(\mathbf{x}_t \,|\, \boldsymbol{\mu}_{t|\tau}^x, \boldsymbol{\Sigma}_{t|\tau}^x)$ of the latent state posterior distributions $p(\mathbf{x}_t|\mathbf{z}_{1:\tau})$. We use the short-hand notation $\mathbf{a}_{b|c}^d$ where $\mathbf{a} = \boldsymbol{\mu}$ denotes the mean $\boldsymbol{\mu}$ and $\mathbf{a} = \boldsymbol{\Sigma}$ denotes the covariance, b denotes the time step under consideration, c denotes the time step up to which we consider measurements, and $d \in \{x, z\}$ denotes either the latent space (x) or the observed space (z).

B. Gaussian RTS Smoothing

Given the filtering distributions $p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \mathcal{N}(\mathbf{x}_t \,|\, \boldsymbol{\mu}_{t|t}^x, \boldsymbol{\Sigma}_{t|t}^x)$, $t = 1, \dots, T$, a sufficient condition for Gaussian smoothing is the computation of Gaussian approximations of the joint distributions $p(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1})$, $t = 1, \dots, T$ [19].

In Gaussian smoothers, the standard smoothing distribution for the dynamic system in (1)–(2) is always

$$p(\mathbf{x}_{t-1}|\mathbf{z}_{1:T}) = \mathcal{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\mu}_{t-1|T}^x, \boldsymbol{\Sigma}_{t-1|T}^x)\,, \quad \text{where} \tag{3}$$

$$\boldsymbol{\mu}_{t-1|T}^x = \boldsymbol{\mu}_{t-1|t-1}^x + \mathbf{J}_{t-1}(\boldsymbol{\mu}_{t|T}^x - \boldsymbol{\mu}_{t|t}^x) \tag{4}$$

$$\boldsymbol{\Sigma}_{t-1|T}^x = \boldsymbol{\Sigma}_{t-1|t-1}^x + \mathbf{J}_{t-1}(\boldsymbol{\Sigma}_{t|T}^x - \boldsymbol{\Sigma}_{t|t}^x)\mathbf{J}_{t-1}^\top \tag{5}$$

$$\mathbf{J}_{t-1} := \boldsymbol{\Sigma}_{t-1,t|t-1}^x(\boldsymbol{\Sigma}_{t|t-1}^x)^{-1} \qquad t = T, \dots, 1\,. \tag{6}$$

Depending on the methodology of computing this joint distribution, we can directly derive arbitrary RTS smoothing algorithms, including the URTSS [16], the EKS [1], [10], the CKS [19], a smoothing extension to the CKF [15], or the GP-URTSS, a smoothing extension to the GP-UKF [12]. The individual smoothers (URTSS, EKS, CKS, GP-based smoothers etc.) simply differ in the way of computing/ estimating the means and covariances required in (4)–(6) [19].

To derive the GP-URTSS, we closely follow the derivation of the URTSS [16]. The GP-URTSS is a novel smoother, but its derivation is relatively straightforward and therefore not detailed in this note. Instead, we detail the derivation of the GP-RTSS, a robust Rauch-Tung-Striebel smoother for GP dynamic systems, which is based on analytic computation of the means and (cross-)covariances in (4)–(6).

In *GP dynamic systems*, the transition function f and the measurement function g in (1)–(2) are modeled by Gaussian processes. This setup is getting more relevant in practical applications such as robotics and control, where it can be difficult to find an accurate parametric form of f and g, respectively [25], [27]. Given the increasing use of GP models in robotics and control, the robustness of Bayesian state estimation is important.

This work was supported in part by ONR MURI under Grant N00014-09-1-1052, by Intel Labs, and by DataPath, Inc.

M. P. Deisenroth is with the TU Darmstadt, Germany, and also with the University of Washington, Seattle, USA. (email: marc@ias.tu-darmstadt.de)

R. D. Turner is with Winton Capital, London, UK, and the University of Cambridge, UK. (email: r.turner@wintoncapital.com)

M. F. Huber is with the AGT Group (R&D) GmbH, Darmstadt, Germany. (email: marco.huber@ieee.org)

U. D. Hanebeck is with the Karlsruhe Institute of Technology (KIT), Germany. (email: Uwe.Hanebeck@ieee.org)

C. E. Rasmussen is with the University of Cambridge, UK, and also with the Max Planck Institute for Biological Cybernetics, Tübingen, Germany. (email: cer54@cam.ac.uk)

II. Gaussian Processes

In the standard GP regression model, we assume that the data $\mathcal{D} := \{\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_n]^\top, \mathbf{y} := [y_1, \dots, y_n]^\top\}$ have been generated according to $y_i = h(\mathbf{x}_i) + \varepsilon_i$, where $h : \mathbb{R}^D \to \mathbb{R}$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is independent (measurement) noise. GPs consider h a random function and infer a posterior distribution over h from data. The posterior is used to make predictions about function values $h(\mathbf{x}_*)$ for arbitrary inputs $\mathbf{x}_* \in \mathbb{R}^D$.

Similar to a Gaussian distribution, which is fully specified by a mean vector and a covariance matrix, a GP is fully specified by a mean *function* $m_h(\,\cdot\,)$ and a covariance *function*

$$\begin{aligned} k_h(\mathbf{x}, \mathbf{x}') &:= \mathbb{E}_h[(h(\mathbf{x}) - m_h(\mathbf{x}))(h(\mathbf{x}') - m_h(\mathbf{x}'))] && (7)\\ &= \mathrm{cov}_h[h(\mathbf{x}), h(\mathbf{x}')] \in \mathbb{R}\,, \quad \mathbf{x},\ \mathbf{x}' \in \mathbb{R}^D && (8) \end{aligned}$$

which specifies the covariance between any two function values. Here, $\mathbb{E}_h$ denotes the expectation with respect to the function h. The covariance function $k_h(\,\cdot\,,\,\cdot\,)$ is also called a *kernel*.

Unless stated otherwise, we consider a prior mean function $m_h \equiv 0$ and use the squared exponential (SE) covariance function with automatic relevance determination

$$k_{\text{SE}}(\mathbf{x}_p, \mathbf{x}_q) := \alpha^2 \exp\left(-\frac{1}{2}(\mathbf{x}_p - \mathbf{x}_q)^\top \mathbf{\Lambda}^{-1}(\mathbf{x}_p - \mathbf{x}_q)\right) \qquad (9)$$

for $\mathbf{x}_p,\ \mathbf{x}_q \in \mathbb{R}^D$, plus a noise covariance function $k_{\text{noise}} := \delta_{pq}\sigma_\varepsilon^2$, such that $k_h = k_{\text{SE}} + k_{\text{noise}}$. The δ denotes the Kronecker symbol that is unity when $p = q$ and zero otherwise, resulting in i.i.d. measurement noise. In (9), $\mathbf{\Lambda} = \mathrm{diag}([\ell_1^2, \dots, \ell_D^2])$ is a diagonal matrix of squared characteristic length-scales ℓ_i, $i = 1, \dots, D$, and α^2 is the signal variance of the latent function h. By using the SE covariance function from (9) we assume that the latent function h is smooth and stationary. Smoothness and stationarity are easier to justify than fixed parametric form of the underlying function.

A. Expressiveness of the Model

Although the SE covariance function k_{SE} and the prior mean function $m_h \equiv 0$ are common defaults, they retain a great deal of expressiveness. Inspired by [20], [28], we demonstrate this expressiveness and show the correspondence of our GP model to a universal function approximator: Consider a function

$$h(x) = \sum\nolimits_{i \in \mathbb{Z}} \lim_{N \to \infty} \frac{1}{N} \sum\nolimits_{n=1}^{N} \gamma_n \exp\left(-\frac{(x - (i + \frac{n}{N}))^2}{\lambda^2}\right) \quad (10)$$

where $\gamma_n \sim \mathcal{N}(0, 1)\,, n = 1, \dots, N$. Note that in the limit, $h(x)$ is represented by infinitely many Gaussian-shaped basis functions along the real axis with width $\lambda/\sqrt{2}$ and prior (Gaussian) random weights γ_n, for $x \in \mathbb{R}$, and for all $i \in \mathbb{Z}$. The model in (10) is considered a universal function approximator. Writing the sums in (10) as an integral over the real axis $\mathbb{R}$, we obtain

$$\begin{aligned} h(x) &= \sum\nolimits_{i \in \mathbb{Z}} \int_i^{i+1} \gamma(s) \exp\left(-\frac{(x - s)^2}{\lambda^2}\right) \mathrm{d}s \\ &= \int_{-\infty}^{\infty} \gamma(s) \exp\left(-\frac{(x - s)^2}{\lambda^2}\right) \mathrm{d}s = (\gamma * \mathcal{K})(x) \qquad (11) \end{aligned}$$

where $\gamma(s) \sim \mathcal{N}(0, 1)$ is a white-noise process and $\mathcal{K}$ is a Gaussian convolution kernel. The function values of h are jointly normal, which follows from the convolution $\gamma * \mathcal{K}$. We now analyze the mean function and the covariance function of h, which fully specify the distribution of h. The only random variables are the weights $\gamma(s)$. Computing the expected function of this model (prior mean function) requires averaging over $\gamma(s)$ and yields

$$\begin{aligned} \mathbb{E}_\gamma[h(x)] &= \int h(x) p(\gamma(s))\, \mathrm{d}\gamma(s) && (12)\\ &\overset{(11)}{=} \int \exp\left(-\frac{(x - s)^2}{\lambda^2}\right) \int \gamma(s) p(\gamma(s))\, \mathrm{d}\gamma(s)\, \mathrm{d}s = 0 && (13) \end{aligned}$$

since $\mathbb{E}_\gamma[\gamma(s)] = 0$. Hence, the mean function of h equals zero everywhere. Let us now find the covariance function. Since the mean function equals zero, for any $x, x' \in \mathbb{R}$ we obtain

$$\begin{aligned} \mathrm{cov}_\gamma[h(x), h(x')] &= \int h(x) h(x') p(\gamma(s))\, \mathrm{d}\gamma(s) \\ &= \int \exp\left(-\frac{(x - s)^2}{\lambda^2}\right) \exp\left(-\frac{(x' - s)^2}{\lambda^2}\right) \\ &\quad \times \int \gamma(s)^2 p(\gamma(s))\, \mathrm{d}\gamma(s)\, \mathrm{d}s \qquad (14) \end{aligned}$$

where we used the definition of h in (11). Using $\mathrm{var}_\gamma[\gamma(s)] = 1$ and completing the squares yields

$$\begin{aligned} \mathrm{cov}_\gamma[h(x), h(x')] &= \int \exp\left(-\frac{2\left(s - \frac{x + x'}{2}\right)^2 + \frac{(x - x')^2}{2}}{\lambda^2}\right) \mathrm{d}s \\ &= \alpha^2 \exp\left(-\frac{(x - x')^2}{2\lambda^2}\right) \qquad (15) \end{aligned}$$

for suitable α^2.

From (13) and (15), we see that the mean function and the covariance function of the universal function approximator in (10) correspond to the GP model assumptions we made earlier: a prior mean function $m_h \equiv 0$ and the SE covariance function in (9) for a one-dimensional input space. Hence, the considered GP prior implicitly assumes latent functions h that can be described by the universal function approximator in (11). Examples of covariance functions that encode different model assumptions are given in [21].

B. Training via Evidence Maximization

For E target dimensions, we train E GPs assuming that the target dimensions are independent at a deterministically given test input (if the test input is uncertain, the target dimensions covary): After observing a data set $\mathcal{D}$, for each (training) target dimension, we learn the $D + 1$ hyper-parameters of the covariance function and the noise variance of the data using *evidence maximization* [20], [21]: Collecting all $(D + 2)E$ hyper-parameters in the vector $\boldsymbol{\theta}$, evidence maximization yields a point estimate $\hat{\boldsymbol{\theta}} \in \mathrm{argmax}_{\boldsymbol{\theta}} \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$. Evidence maximization automatically trades off data fit with function complexity and avoids overfitting [21].

From here onward, we consider the GP dynamic system setup, where two GP models have been trained using evidence maximization: $\mathcal{GP}_f$, which models the mapping $\mathbf{x}_{t-1} \mapsto \mathbf{x}_t$, $\mathbb{R}^D \to \mathbb{R}^D$, see (1), and $\mathcal{GP}_g$, which models the mapping $\mathbf{x}_t \mapsto \mathbf{z}_t$, $\mathbb{R}^D \to \mathbb{R}^E$, see (2). To keep the notation uncluttered, we do not explicitly condition on the hyper-parameters $\hat{\boldsymbol{\theta}}$ and the training data $\mathcal{D}$ in the following.

III. Robust Smoothing in Gaussian Process Dynamic Systems

Analytic moment-based filtering in GP dynamic systems has been proposed in [13], where the filter distribution is given by

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{z}_{1:t}) &= \mathcal{N}(\mathbf{x}_t \mid \boldsymbol{\mu}_{t|t}^x, \mathbf{\Sigma}_{t|t}^x) && (16)\\ \boldsymbol{\mu}_{t|t}^x &= \boldsymbol{\mu}_{t|t-1}^x + \mathbf{\Sigma}_{t|t-1}^{xz} \left(\mathbf{\Sigma}_{t|t-1}^z\right)^{-1} (\mathbf{z}_t - \boldsymbol{\mu}_{t|t-1}^z) && (17)\\ \mathbf{\Sigma}_{t|t}^x &= \mathbf{\Sigma}_{t|t-1}^x - \mathbf{\Sigma}_{t|t-1}^{xz} \left(\mathbf{\Sigma}_{t|t-1}^z\right)^{-1} \mathbf{\Sigma}_{t|t-1}^{zx} && (18) \end{aligned}$$

for $t = 1, \ldots, T$. Here, we extend these filtering results to analytic moment-based smoothing, where we explicitly take nonlinearities into account (no linearization required) while propagating full Gaussian densities (no sigma/cubature-point representation required) through nonlinear GP models.

In the following, we detail our novel RTS smoothing approach for GP dynamic systems. We fit our smoother in the standard frame of (4)–(6). For this, we compute the means and covariances of the Gaussian approximation

$$\mathcal{N}\left(\begin{bmatrix}\mathbf{x}_{t-1}\\ \mathbf{x}_t\end{bmatrix}\middle|\begin{bmatrix}\boldsymbol{\mu}^x_{t-1|t-1}\\ \boldsymbol{\mu}^x_{t|t-1}\end{bmatrix},\begin{bmatrix}\boldsymbol{\Sigma}^x_{t-1|t-1} & \boldsymbol{\Sigma}^x_{t-1,t|t-1}\\ \boldsymbol{\Sigma}^x_{t,t-1|t-1} & \boldsymbol{\Sigma}^x_{t|t-1}\end{bmatrix}\right) \tag{19}$$

to the joint $p(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1})$, after which the smoother is fully determined [19]. Our approximation does not involve sampling, linearization, or numerical integration. Instead, we present closed-form expressions of a deterministic Gaussian approximation of the joint distribution in (19).

In our case, the mapping $\mathbf{x}_{t-1} \mapsto \mathbf{x}_t$ is not known, but instead it is distributed according to $\mathcal{GP}_f$, a distribution over system functions. For robust filtering and smoothing, we therefore need to take the GP (model) uncertainty into account by Bayesian averaging according to the GP distribution [13], [29]. The marginal $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) = \mathcal{N}(\boldsymbol{\mu}^x_{t-1|t-1}, \boldsymbol{\Sigma}^x_{t-1|t-1})$ is known from filtering [13]. In Section III-A, we compute the mean and covariance of second marginal $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ and then in Section III-B the cross-covariance terms $\boldsymbol{\Sigma}^x_{t-1,t|t-1} = \mathrm{cov}[\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1}]$.

A. Marginal Distribution

1) Marginal Mean: Using the system equation (1) and integrating over all three sources of uncertainties (the system noise, the state $\mathbf{x}_{t-1}$, and the system function itself), we apply the law of total expectation and obtain the marginal mean

$$\boldsymbol{\mu}^x_{t|t-1} = \mathbb{E}_{\mathbf{x}_{t-1}}\left[\mathbb{E}_f[f(\mathbf{x}_{t-1})|\mathbf{x}_{t-1}]|\mathbf{z}_{1:t-1}\right]. \tag{20}$$

The expectations in (20) are taken with respect to the posterior GP distribution $p(f)$ and the filter distribution $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) = \mathcal{N}(\boldsymbol{\mu}^x_{t-1|t-1}, \boldsymbol{\Sigma}^x_{t-1|t-1})$ at time step $t-1$. Equation (20) can be rewritten as $\boldsymbol{\mu}^x_{t|t-1} = \mathbb{E}_{\mathbf{x}_{t-1}}[m_f(\mathbf{x}_{t-1})|\mathbf{z}_{1:t-1}]$ where $m_f(\mathbf{x}_{t-1}) :$ $= \mathbb{E}_f[f(\mathbf{x}_{t-1})|\mathbf{x}_{t-1}]$ is the posterior mean function of $\mathcal{GP}_f$. Writing m_f as a finite sum over the SE kernels centered at all n training inputs [21], the predicted mean for each target dimension $a = 1, \ldots, D$ is

$$\begin{aligned}\left(\boldsymbol{\mu}^x_{t|t-1}\right)_a &= \int m_{f_a}(\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\,\mathrm{d}\mathbf{x}_{t-1} \qquad (21)\\ &= \sum\nolimits_{i=1}^{n}\beta^x_{a_i}\int k_{f_a}(\mathbf{x}_{t-1}, \mathbf{x}_i)p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\,\mathrm{d}\mathbf{x}_{t-1}\end{aligned}$$

where $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) = \mathcal{N}(\mathbf{x}_{t-1}\,|\,\boldsymbol{\mu}^x_{t-1|t-1}, \boldsymbol{\Sigma}^x_{t-1|t-1})$ is the filter distribution at time $t-1$. Moreover, $\mathbf{x}_i$, $i = 1, \ldots, n$, are the training set of $\mathcal{GP}_f$, k_{f_a} is the covariance function of $\mathcal{GP}_f$ for the ath target dimension (GP hyper-parameters are not shared across dimensions), and $\boldsymbol{\beta}^x_a := (\mathbf{K}_{f_a} + \sigma^2_{w_a}\mathbf{I})^{-1}\mathbf{y}_a \in \mathbb{R}^n$. For dimension a, $\mathbf{K}_{f_a}$ denotes the kernel matrix (Gram matrix), where $\mathbf{K}_{f_a}(i,j) = k_{f_a}(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \ldots, n$. Moreover, $\mathbf{y}_a$ are the training targets, and $\sigma^2_{w_a}$ is the learned system noise variance. The vector $\boldsymbol{\beta}^x_a$ has been pulled out of the integration since it is independent of $\mathbf{x}_{t-1}$. Note that $\mathbf{x}_{t-1}$ serves as a test input from the perspective of the GP regression model.

For the SE covariance function in (9), the integral in (21) can be computed analytically (other tractable choices are covariance functions containing combinations of squared exponentials, trigonometric functions, and polynomials). The marginal mean is given as

$$\left(\boldsymbol{\mu}^x_{t|t-1}\right)_a = (\boldsymbol{\beta}^x_a)^\top\mathbf{q}^x_a \tag{22}$$

where we defined

$$\begin{aligned}q^x_{a_i} &:= \alpha^2_{f_a}|\boldsymbol{\Sigma}^x_{t-1|t-1}\boldsymbol{\Lambda}^{-1}_a + \mathbf{I}|^{-\frac{1}{2}}\\ &\quad\times\exp\Big(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}^x_{t-1|t-1})^\top\\ &\quad\times\mathbf{S}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}^x_{t-1|t-1})\Big) \qquad (23)\\ \mathbf{S} &:= \boldsymbol{\Sigma}^x_{t-1|t-1} + \boldsymbol{\Lambda}_a \qquad (24)\end{aligned}$$

$i = 1, \ldots, n$, being the solution to the integral in (21). Here, $\alpha^2_{f_a}$ is the signal variance of the ath target dimension of $\mathcal{GP}_f$, a learned hyper-parameter of the SE covariance function, see (9).

2) Marginal Covariance Matrix: We now explicitly compute the entries of the corresponding covariance $\boldsymbol{\Sigma}^x_{t|t-1}$. Using the law of total covariance, we obtain for $a, b = 1, \ldots, D$

$$\begin{aligned}(\Sigma^x_{t|t-1})_{(ab)} &= \mathrm{cov}_{\mathbf{x}_{t-1},f,\mathbf{w}}\left[x^{(a)}_t, x^{(b)}_t|\mathbf{z}_{1:t-1}\right]\\ &= \mathbb{E}_{\mathbf{x}_{t-1}}\big[\mathrm{cov}_{f,\mathbf{w}}[f_a(\mathbf{x}_{t-1}) + w_a, f_b(\mathbf{x}_{t-1})\\ &\qquad + w_b|\mathbf{x}_{t-1}]|\mathbf{z}_{1:t-1}\big]\\ &\quad + \mathrm{cov}_{\mathbf{x}_{t-1}}\big[\mathbb{E}_{f_a}[f_a(\mathbf{x}_{t-1})|\mathbf{x}_{t-1}],\\ &\qquad \mathbb{E}_{f_b}[f_b(\mathbf{x}_{t-1})|\mathbf{x}_{t-1}]|\mathbf{z}_{1:t-1}\big] \qquad (25)\end{aligned}$$

where we exploited in the last term that the system noise $\mathbf{w}$ has mean zero. Note that (25) is the sum of the covariance of (conditional) expected values and the expectation of a (conditional) covariance. We analyze these terms in the following.

The *covariance of the expectations* in (25) is

$$\int m_{f_a}(\mathbf{x}_{t-1})m_{f_b}(\mathbf{x}_{t-1})p(\mathbf{x}_{t-1})\,\mathrm{d}\mathbf{x}_{t-1} - (\mu^x_{t|t-1})_a(\mu^x_{t|t-1})_b \tag{26}$$

where we used that $\mathbb{E}_f[f(\mathbf{x}_{t-1})|\mathbf{x}_{t-1}] = m_f(\mathbf{x}_{t-1})$. With $\boldsymbol{\beta}^x_a = (\mathbf{K}_a + \sigma^2_{w_a}\mathbf{I})^{-1}\mathbf{y}_a$ and $m_{f_a}(\mathbf{x}_{t-1}) = k_{f_a}(\mathbf{X}, \mathbf{x}_{t-1})^\top\boldsymbol{\beta}^x_a$, we obtain

$$\begin{aligned}&\mathrm{cov}_{\mathbf{x}_{t-1}}[m_{f_a}(\mathbf{x}_{t-1})\,, m_{f_b}(\mathbf{x}_{t-1})|\mathbf{z}_{1:t-1}]\\ &\qquad = (\boldsymbol{\beta}^x_a)^\top\mathbf{Q}\boldsymbol{\beta}^x_b - (\mu^x_{t|t-1})_a(\mu^x_{t|t-1})_b\,. \qquad (27)\end{aligned}$$

Following [30], the entries of $\mathbf{Q} \in \mathbb{R}^{n\times n}$ are given as

$$\begin{aligned}Q_{ij} &= k_{f_a}(\mathbf{x}_i, \boldsymbol{\mu}^x_{t-1|t-1})k_{f_b}(\mathbf{x}_j, \boldsymbol{\mu}^x_{t-1|t-1})/\sqrt{|\mathbf{R}|}\\ &\quad\times\exp\left(\frac{1}{2}\mathbf{z}^\top_{ij}\mathbf{R}^{-1}\boldsymbol{\Sigma}^x_{t-1|t-1}\mathbf{z}_{ij}\right) = \exp(n^2_{ij})/\sqrt{|\mathbf{R}|} \qquad (28)\\ n^2_{ij} &= \log(\alpha^2_{f_a}) + \log(\alpha^2_{f_b})\\ &\quad - \frac{1}{2}\left(\boldsymbol{\zeta}^\top_i\boldsymbol{\Lambda}^{-1}_a\boldsymbol{\zeta}_i + \boldsymbol{\zeta}^\top_j\boldsymbol{\Lambda}^{-1}_b\boldsymbol{\zeta}_j - \mathbf{z}^\top_{ij}\mathbf{R}^{-1}\boldsymbol{\Sigma}^x_{t-1|t-1}\mathbf{z}_{ij}\right)\end{aligned}$$

where we defined $\mathbf{R} := \boldsymbol{\Sigma}^x_{t-1|t-1}(\boldsymbol{\Lambda}^{-1}_a + \boldsymbol{\Lambda}^{-1}_b) + \mathbf{I}$, $\boldsymbol{\zeta}_i := \mathbf{x}_i - \boldsymbol{\mu}^x_{t-1|t-1}$, and $\mathbf{z}_{ij} := \boldsymbol{\Lambda}^{-1}_a\boldsymbol{\zeta}_i + \boldsymbol{\Lambda}^{-1}_b\boldsymbol{\zeta}_j$.

The *expected covariance* in (25) is given as

$$\mathbb{E}_{\mathbf{x}_{t-1}}\left[\mathrm{cov}_f[f_a(\mathbf{x}_{t-1}), f_b(\mathbf{x}_{t-1})|\mathbf{x}_{t-1}]|\mathbf{z}_{1:t-1}\right] + \delta_{ab}\sigma^2_{w_a} \tag{29}$$

since the noise covariance matrix $\boldsymbol{\Sigma}_w$ is diagonal. Following our GP training assumption that different target dimensions do not covary if the input is deterministically given, (29) is only non-zero if $a = b$, i.e., (29) plays a role only for diagonal entries of $\boldsymbol{\Sigma}^x_{t|t-1}$. For these diagonal entries ($a = b$), the expected covariance in (29) is

$$\alpha^2_{f_a} - \mathrm{tr}\left((\mathbf{K}_{f_a} + \sigma^2_{w_a}\mathbf{I})^{-1}\mathbf{Q}\right) + \sigma^2_{w_a}\,. \tag{30}$$

Hence, the desired marginal covariance matrix in (25) is

$$(\Sigma^x_{t|t-1})_{ab} = \left\{\begin{array}{rl}\text{Eq. (27)} + \text{Eq. (30)}\,, & \text{if } a = b\\ \text{Eq. (27)}\,, & \text{otherwise}\end{array}\right. \tag{31}$$

We have now solved for the marginal distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ in (19). Since the approximate Gaussian filter distribution

$p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) = \mathcal{N}(\boldsymbol{\mu}^x_{t-1|t-1}, \boldsymbol{\Sigma}^x_{t-1|t-1})$ is also known, it remains to compute the cross-covariance $\boldsymbol{\Sigma}^x_{t-1,t|t-1}$ to fully determine the Gaussian approximation in (19).

B. Cross-Covariance

By the definition of a covariance and the system equation (1), the missing cross-covariance matrix $\boldsymbol{\Sigma}^x_{t-1,t|t-1}$ in (19) is

$$\begin{aligned}\boldsymbol{\Sigma}^x_{t-1,t|t-1} = \mathbb{E}_{\mathbf{x}_{t-1},f,\mathbf{w}_t}&\left[\mathbf{x}_{t-1}\left(f(\mathbf{x}_{t-1})+\mathbf{w}_t\right)^\top|\mathbf{z}_{1:t-1}\right]\\ &- \boldsymbol{\mu}^x_{t-1|t-1}(\boldsymbol{\mu}^x_{t|t-1})^\top \qquad (32)\end{aligned}$$

where $\boldsymbol{\mu}^x_{t-1|t-1}$ is the mean of the filter update at time step $t-1$ and $\boldsymbol{\mu}^x_{t|t-1}$ is the mean of the time update, see (20). Note that we explicitly average out the model uncertainty about f. Using the law of total expectations, we obtain

$$\begin{aligned}\boldsymbol{\Sigma}^x_{t-1,t|t-1} &= \mathbb{E}_{\mathbf{x}_{t-1}}\left[\mathbf{x}_{t-1}\,\mathbb{E}_{f,\mathbf{w}_t}[f(\mathbf{x}_{t-1})+\mathbf{w}_t|\mathbf{x}_{t-1}]^\top|\mathbf{z}_{1:t-1}\right]\\ &\qquad - \boldsymbol{\mu}^x_{t-1|t-1}(\boldsymbol{\mu}^x_{t|t-1})^\top \qquad (33)\\ &= \mathbb{E}_{\mathbf{x}_{t-1}}\left[\mathbf{x}_{t-1}m_f(\mathbf{x}_{t-1})^\top|\mathbf{z}_{1:t-1}\right]\\ &\qquad - \boldsymbol{\mu}^x_{t-1|t-1}(\boldsymbol{\mu}^x_{t|t-1})^\top \qquad (34)\end{aligned}$$

where we used the fact that $\mathbb{E}_{f,\mathbf{w}_t}[f(\mathbf{x}_{t-1})+\mathbf{w}_t|\mathbf{x}_{t-1}] = m_f(\mathbf{x}_{t-1})$ is the mean function of $\mathcal{GP}_f$, which models the mapping $\mathbf{x}_{t-1}\mapsto\mathbf{x}_t$, evaluated at $\mathbf{x}_{t-1}$. We thus obtain

$$\begin{aligned}\boldsymbol{\Sigma}^x_{t-1,t|t-1} = \int &\mathbf{x}_{t-1}m_f(\mathbf{x}_{t-1})^\top p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\,\mathrm{d}\mathbf{x}_{t-1}\\ &- \boldsymbol{\mu}^x_{t-1|t-1}(\boldsymbol{\mu}^x_{t|t-1})^\top\,. \qquad (35)\end{aligned}$$

Writing $m_f(\mathbf{x}_{t-1})$ as a finite sum over kernels [21] and moving the integration into this sum, the integration in (35) turns into

$$\begin{aligned}&\int \mathbf{x}_{t-1}m_{f_a}(\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\,\mathrm{d}\mathbf{x}_{t-1}\\ &= \sum_{i=1}^n \beta^x_{a_i}\int \mathbf{x}_{t-1}k_{f_a}(\mathbf{x}_{t-1},\mathbf{x}_i)p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\,\mathrm{d}\mathbf{x}_{t-1}\end{aligned}$$

for each state dimension $a=1,\dots,D$. With the SE covariance function k_{SE} defined in (9), we compute the integral analytically and obtain

$$\begin{aligned}&\int \mathbf{x}_{t-1}m_{f_a}(\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\,\mathrm{d}\mathbf{x}_{t-1} \qquad (36)\\ &= \sum_{i=1}^n \beta^x_{a_i}\int \mathbf{x}_{t-1}c_3\mathcal{N}(\mathbf{x}_i,\boldsymbol{\Lambda}_a)\mathcal{N}(\boldsymbol{\mu}^x_{t-1|t-1},\boldsymbol{\Sigma}^x_{t-1|t-1})\,\mathrm{d}\mathbf{x}_{t-1}\end{aligned}$$

where we defined $c_3^{-1} = (\alpha^2_{f_a}(2\pi)^{\frac{D}{2}}\sqrt{|\boldsymbol{\Lambda}_a|})^{-1}$, such that $k_{f_a}(\mathbf{x}_{t-1},\mathbf{x}_i) = c_3\mathcal{N}(\mathbf{x}_{t-1}\,|\,\mathbf{x}_i,\boldsymbol{\Lambda}_a)$. In the definition of c_3, $\alpha^2_{f_a}$ is a hyper-parameter of $\mathcal{GP}_f$ responsible for the variance of the latent function in dimension a. Using the definition of $\mathbf{S}$ in (24), the product of the two Gaussians in (36) results in a new (unnormalized) Gaussian $c_4^{-1}\mathcal{N}(\mathbf{x}_{t-1}\,|\,\boldsymbol{\psi}_i,\boldsymbol{\Psi})$ with

$$\begin{aligned}c_4^{-1} &= (2\pi)^{-\frac{D}{2}}|\boldsymbol{\Lambda}_a+\boldsymbol{\Sigma}^x_{t-1|t-1}|^{-\frac{1}{2}}\\ &\quad\times\exp\left(-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu}^x_{t-1|t-1})^\top\mathbf{S}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}^x_{t-1|t-1})\right)\\ \boldsymbol{\Psi} &= \left(\boldsymbol{\Lambda}_a^{-1}+(\boldsymbol{\Sigma}^x_{t-1|t-1})^{-1}\right)^{-1}\\ \boldsymbol{\psi}_i &= \boldsymbol{\Psi}\left(\boldsymbol{\Lambda}_a^{-1}\mathbf{x}_i+(\boldsymbol{\Sigma}^x_{t-1|t-1})^{-1}\boldsymbol{\mu}^x_{t-1|t-1}\right)\,.\end{aligned}$$

Pulling all constants outside the integral in (36), the integral determines the expected value of the product of the two Gaussians, $\boldsymbol{\psi}_i$. For $a=1,\dots,D$, we obtain

$$\mathbb{E}[\mathbf{x}_{t-1}\,x_{t_a}|\mathbf{z}_{1:t-1}] = \sum\nolimits_{i=1}^n c_3c_4^{-1}\beta^x_{a_i}\boldsymbol{\psi}_i\,.$$

Using $c_3c_4^{-1} = q^x_{a_i}$, see (23), and some matrix identities, we finally obtain

$$\sum\nolimits_{i=1}^n \beta^x_{a_i}q^x_{a_i}\boldsymbol{\Sigma}^x_{t-1|t-1}(\boldsymbol{\Sigma}^x_{t-1|t-1}+\boldsymbol{\Lambda}_a)^{-1}(\mathbf{x}_i-\boldsymbol{\mu}^x_{t-1|t-1}) \qquad (37)$$

for $\text{cov}_{\mathbf{x}_{t-1},f,\mathbf{w}_t}[\mathbf{x}_{t-1},x_{t_a}|\mathbf{z}_{1:t-1}]$ and the joint covariance matrix of $p(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{z}_{1:t-1})$ and, hence, the full Gaussian approximation in (19) is completely determined.

With the mean and the covariance of the joint distribution $p(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{z}_{1:t-1})$ given by (22), (31), (37), and the filter step, all necessary components are provided to compute the smoothing distribution $p(\mathbf{x}_t|\mathbf{z}_{1:T})$ analytically [19].

IV. Simulations

In the following, we present results analyzing the robustness of state-of-the art nonlinear filters (Section IV-A) and the performance of the corresponding smoothers (Section IV-B).

A. Filter Robustness

We considered the nonlinear stochastic dynamic system

$$x_t = \frac{x_{t-1}}{2}+\frac{25\,x_{t-1}}{1+x^2_{t-1}}+w_t\,,\quad w_t\sim\mathcal{N}(0,\sigma^2_w=0.2^2) \qquad (38)$$

$$z_t = 5\,\sin(x_t)+v_t\,,\quad v_t\sim\mathcal{N}(0,\sigma^2_v=0.2^2) \qquad (39)$$

which is a modified version of the model used in [18], [31]. The system was modified in two ways: First, (38) does not contain a purely time-dependent term in the system, which would not allow for learning stationary transition dynamics. Second, we substituted a sinusoidal measurement function for the originally quadratic measurement function used by [18] and [31]. The sinusoidal measurement function increases the difficulty in computing the marginal distribution $p(\mathbf{z}_t|\mathbf{z}_{1:t-1})$ if the time update distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ is fairly uncertain: While the quadratic measurement function can only lead to bimodal distributions (assuming a Gaussian input distribution), the sinusoidal measurement function in (39) can lead to an arbitrary number of modes—for a broad input distribution.

The prior variance was set to $\sigma^2_0=0.5^2$, i.e., the initial uncertainty was fairly high. The system and measurement noises (see (38)–(39)) were relatively small considering the amplitudes of the system function and the measurement function. For the numerical analysis, a linear grid in the interval $[-3,3]$ of mean values $(\mu^x_0)_i$, $i=1,\dots,100$, was defined. Then, a single latent (initial) state $x^{(i)}_0$ was sampled from $p(x^{(i)}_0)=\mathcal{N}((\mu^x_0)_i,\sigma^2_0)$, $i=1,\dots,100$.

For the dynamic system in (38)–(39), we analyzed the robustness in a single filter step of the EKF, the UKF, the CKF, and a SIR PF (sequential importance resampling particle filter) with 200 particles, the GP-UKF, and the GP-ADF against the ground truth, closely approximated by the Gibbs-filter [19]. Compared to the evaluation of longer trajectories, evaluating a single filter step makes it easier to analyze the robustness of individual filtering algorithms.

Table I summarizes the expected performances (RMSE: root-mean-square error, MAE: mean-absolute error, NLL: negative log-likelihood) of the EKF, the UKF, the CKF, the GP-UKF, the GP-ADF, the Gibbs-filter, and the SIR PF for estimating the latent state x. The results in the table are based on averages over 1,000 test runs and 100 randomly sampled start states per test run (see experimental setup). The table also reports the 95% standard error of the expected performances. Table I indicates that the GP-ADF was the most robust filter and statistically significantly outperformed all filters but the sampling-based Gibbs-filter and the SIR PF. The green color highlights a near-optimal Gaussian filter (Gibbs-filter) and the near-optimal particle filter. Amongst all other filters the GP-ADF was

TABLE I

AVERAGE FILTER PERFORMANCES (RMSE, MAE, NLL) WITH STANDARD ERRORS (95% CONFIDENCE INTERVAL) AND P-VALUES TESTING THE HYPOTHESIS THAT THE OTHER FILTERS ARE BETTER THAN THE GP-ADF USING A ONE-SIDED T-TEST.

	RMSE_x (p-value)	MAE_x (p-value)	NLL_x (p-value)
EKF [10]	3.62 ± 0.212 ($p = 4.1 \times 10^{-2}$)	2.36 ± 0.176 ($p = 0.38$)	$3.05 \times 10^3 \pm 3.02 \times 10^2$ ($p < 10^{-4}$)
UKF [11]	10.5 ± 1.08 ($p < 10^{-4}$)	8.58 ± 0.915 ($p < 10^{-4}$)	25.6 ± 3.39 ($p < 10^{-4}$)
CKF [15]	9.24 ± 1.13 ($p = 2.8 \times 10^{-4}$)	7.31 ± 0.941 ($p = 4.2 \times 10^{-4}$)	$2.22 \times 10^2 \pm 17.5$ ($p < 10^{-4}$)
GP-UKF [12]	5.36 ± 0.461 ($p = 7.9 \times 10^{-4}$)	3.84 ± 0.352 ($p = 3.3 \times 10^{-3}$)	6.02 ± 0.497 ($p < 10^{-4}$)
GP-ADF [13]	$\mathbf{2.85 \pm 0.174}$	$\mathbf{2.17 \pm 0.151}$	$\mathbf{1.97 \pm 6.55 \times 10^{-2}}$
Gibbs-filter [19]	$\mathbf{2.82 \pm 0.171}$ ($p = 0.54$)	$\mathbf{2.12 \pm 0.148}$ ($p = 0.56$)	$\mathbf{1.96 \pm 6.62 \times 10^{-2}}$ ($p = 0.55$)
SIR PF	$\mathbf{1.57 \pm 7.66 \times 10^{-2}}$ ($p = 1.0$)	$\mathbf{0.36 \pm 2.28 \times 10^{-2}}$ ($p = 1.0$)	$\mathbf{1.03 \pm 7.30 \times 10^{-2}}$ ($p = 1.0$)

the closest Gaussian filter to the computationally expensive Gibbs-filter [19]. Note that the SIR PF is not a Gaussian filter and is able to express multi-modality in distributions. Therefore, its performance is typically better than the one of Gaussian filters. The difference between the SIR PF and a near-optimal Gaussian filter, the Gibbs-filter, is expressed in Table I. The performance difference essentially depicts how much we lost by using a Gaussian filter instead of a particle filter. The NLL values for the SIR PF were obtained by moment-matching the particles.

The poor performance of the EKF was due to linearization errors. The filters based on small sample approximations of densities (UKF, GP-UKF, CKF) suffered from the degeneracy of these approximations, which is illustrated in Figure 1. Note that the CKF uses a smaller set of cubature points than the UKF to determine predictive distributions, which makes the CKF statistically even less robust than the UKF.

B. Smoother Robustness

We considered a pendulum tracking example taken from [13]. We evaluated the performances of four filters and smoothers, the EKF/EKS, the UKF/URTSS, the GP-UKF/GP-URTSS, the CKF/CKS, the Gibbs-filter, and the GP-ADF/GP-RTSS. The pendulum had mass $m = 1\,\text{kg}$ and length $l = 1\,\text{m}$. The state $\mathbf{x} = [\dot\varphi, \varphi]^\top$ of the pendulum was given by the angle φ (measured anti-clockwise from hanging down) and the angular velocity $\dot\varphi$. The pendulum could exert a constrained torque $u \in [-5, 5]\,\text{Nm}$. We assumed a frictionless system such that the transition function f was

$$f(\mathbf{x}_t, u_t) = \int_t^{t+\Delta_t} \begin{bmatrix} \dfrac{u(\tau) - 0.5\,mlg\sin\varphi(\tau)}{0.25\,ml^2 + I} \\ \dot\varphi(\tau) \end{bmatrix} \mathrm{d}\tau \tag{40}$$

where I is the moment of inertia and g the acceleration of gravity. Then, the successor state

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+\Delta_t} = f(\mathbf{x}_t, u_t) + \mathbf{w}_t \tag{41}$$

was computed using an ODE solver for (40) with a zero-order hold control signal $u(\tau)$. In (41), we set $\boldsymbol{\Sigma}_w = \text{diag}([0.5^2, 0.1^2])$. In our experiment, the torque was sampled randomly according to $u \sim \mathcal{U}[-5, 5]\,\text{Nm}$ and implemented using a zero-order-hold controller. Every time increment $\Delta_t = 0.2\,\text{s}$, the state was measured according to

$$z_t = \arctan\left(\frac{-1 - l\sin(\varphi_t)}{0.5 - l\cos(\varphi_t)}\right) + v_t\,, \quad \sigma_v^2 = 0.05^2\,. \tag{42}$$

Note that the scalar measurement equation (42) solely depends on the angle. Thus, the full distribution of the latent state $\mathbf{x}$ had to be inferred using the cross-correlation information between the angle and the angular velocity.

Trajectories of length $T = 6\,\text{s} = 30$ time steps were started from a state sampled from the prior $p(\mathbf{x}_0) = \mathcal{N}(\boldsymbol{\mu}_0^x, \boldsymbol{\Sigma}_0^x)$ with $\boldsymbol{\mu}_0^x = [0, 0]^\top$ and $\boldsymbol{\Sigma}_0^x = \text{diag}([0.01^2, (\frac{\pi}{16})^2])$. For each trajectory, GP models $\mathcal{GP}_f$ and $\mathcal{GP}_g$ are learned based on randomly generated data using either 250 or 20 data points.

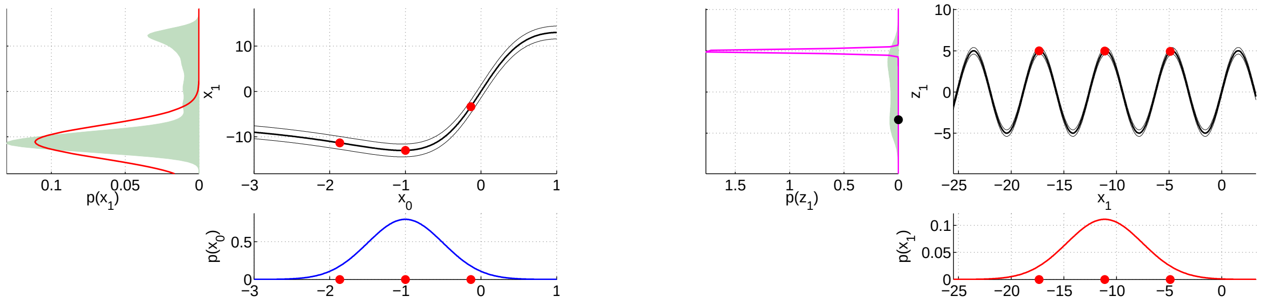
Table II reports the expected values of the NLL_x-measure for the EKF/EKS, the UKF/URTSS, the GP-UKF/GP-URTSS, the GP-ADF/GP-RTSS, and the CKF/CKS when tracking the pendulum over a horizon of 6 s, averaged over 1,000 runs. The $\star$ indicates a method developed in this paper. As in the example in Section IV-A, the NLL_x-measure emphasizes the robustness of our proposed method: The GP-RTSS is the only method that consistently reduced the negative log-likelihood value compared to the corresponding filtering algorithm. Increasing the NLL_x-values (red color in Table II) occurred when the filter distribution could not explain the latent state/measurement, an example of which is given in Figure 1(b). Even with only 20 training points, the GP-ADF/GP-RTSS outperformed the commonly used EKF/EKS, UKF/URTSS, CKF/CKS.

We experimented with even smaller signal-to-noise ratios. The GP-RTSS remained robust, while the other smoothers remained unstable.

V. DISCUSSION AND CONCLUSION

In this paper, we presented the GP-RTSS, an analytic Rauch-Tung-Striebel smoother for GP dynamic systems, where the GPs with SE covariance functions are practical implementations of universal function approximators. We showed that the GP-RTSS is more robust to nonlinearities than state-of-the-art smoothers. There are two main reasons for this: First, the GP-RTSS relies neither on linearization (EKS) nor on density approximations (URTSS/CKS) to compute an optimal Gaussian approximation of the predictive distribution when mapping a Gaussian distribution through a nonlinear function. This property avoids incoherent estimates of the filtering and smoothing distributions as discussed in Sec IV-A. Second, GPs allow for more robust "system identification" than standard methods since they coherently represent uncertainties about the system and measurement functions at locations that have not been encountered in the data collection phase. The GP-RTSS is a robust smoother since it accounts for model uncertainties in a principled Bayesian way.

After training the GPs, which can be performed offline, the computational complexity of the GP-RTSS (including filtering) is $\mathcal{O}(T(E^3 + n^2(D^3 + E^3)))$ for a time series of length T. Here, n is the size of the GP training sets, and D and E are the dimensions of the state and the measurements, respectively. The computational complexity is due to the inversion of the D and E-dimensional covariance matrices, and the computation of the matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$ in (28), required for each entry of a D and E-dimensional covariance matrix. The computational complexity scales linearly with the number of time steps. The computational demand of classical Gaussian smoothers, such as the URTSS and the EKS is $\mathcal{O}(T(D^3 + E^3))$. Although not reported here, we verified the computational complexity experimentally. Approximating the online computations of the GP-RTSS by numerical integration or grids scales poorly with increasing dimension. These problems already appear in the histogram filter [3]. By explicitly providing equations for the solution of the involved



(a) (b)

Fig. 1. Degeneracy of the unscented transformation (UT) underlying the UKF. Input distributions to the UT are the Gaussians in the subfigures at the bottom in each panel. The functions the UT is applied to are shown in the top right subfigures, i.e, the transition mapping (38), in Panel (a), and the measurement mapping (39), in Panel (b). Sigma points are marked by red dots. The predictive distributions are shown in the left subfigures of each panel. The true predictive distributions are the shaded areas; the UT predictive distributions are the solid Gaussians. The predictive distribution of the time update in Panel (a) equals the input distribution at the bottom of Panel (b). (a) UKF time update $p(x_1|\emptyset)$, which misses out substantial probability mass of the true predictive distribution; UKF determines $p(z_1|\emptyset)$, which is too sensitive and cannot explain the actual measurement z_1 (black dot, left subfigure).

TABLE II

Expected filtering and smoothing performances (pendulum tracking) with 95% confidence intervals.

Filters	EKF [10]	UKF [11]	CKF [15]	GP-UKF$_{250}$ [12]	GP-ADF$_{250}$ [13]	GP-ADF$_{20}$ [13]
$\mathbb{E}[\mathrm{NLL}_x]$	$1.6 \times 10^2 \pm 29.1$	6.0 ± 3.02	28.5 ± 9.83	4.4 ± 1.32	$\mathbf{1.44 \pm 0.117}$	6.63 ± 0.149
Smoothers	EKS [10]	URTSS [16]	CKS [19]	GP-URTSS$^\star_{250}$	GP-RTSS$^\star_{250}$	GP-RTSS$^\star_{20}$
$\mathbb{E}[\mathrm{NLL}_x]$	$\mathbf{3.3 \times 10^2 \pm 60.5}$	$\mathbf{17.2 \pm 10.0}$	$\mathbf{72.0 \pm 25.1}$	$\mathbf{10.3 \pm 3.85}$	$\mathbf{1.04 \pm 0.204}$	6.57 ± 0.148

integrals, we show that numerical integration is not necessary and the GP-RTSS is a practical approach to filtering in GP dynamic systems.

Although the GP-RTSS is computationally more involved than the URTSS, the EKS, and the CKS, this does not necessarily imply that smoothing with the GP-RTSS is slower: function evaluations, which are heavily used by the EKS/CKS/URTSS, are not necessary in the GP-RTSS (after training). In the pendulum example, repeatedly calling the ODE solver caused the EKS/CKS/URTSS to be slower than the GP-RTSS (with 250 training points) by a factor of two.

The increasing use of GPs for model learning in robotics and control will eventually require principled smoothing methods for GP dynamic systems. To our best knowledge, the proposed GP-RTSS is the most principled GP-smoother since all computations can be performed analytically without function linearization or sigma/cubature point representation of densities, while exactly integrating out the model uncertainty induced by the GP distribution.

Matlab code is publicly available at `http://mloss.org`.

References

[1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Dover Publications, 2005.

[2] K. J. Åström, *Introduction to Stochastic Control Theory*. Dover Publications, Inc., 2006.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[4] S. T. Roweis and Z. Ghahramani, *Kalman Filtering and Neural Networks*. Wiley, 2001, ch. Learning Nonlinear Dynamical Systems using the EM Algorithm, pp. 175–220.

[5] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[6] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum Likelihood Estimates of Linear Dynamical Systems," *AIAA Journal*, vol. 3, pp. 1445–1450, 1965.

[7] S. Roweis and Z. Ghahramani, "A Unifying Review of Linear Gaussian Models," *Neural Computation*, vol. 11, no. 2, pp. 305–345, 1999.

[8] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, 2001.

[9] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[10] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press, Inc., 1979, vol. 141.

[11] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[12] J. Ko and D. Fox, "GP-BayesFilters: Bayesian Filtering using Gaussian Process Prediction and Observation Models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[13] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic Moment-based Gaussian Process Filtering," in *International Conference on Machine Learning*, 2009, pp. 225–232.

[14] U. D. Hanebeck, "Optimal Filtering of Nonlinear Systems Based on Pseudo Gaussian Densities," in *Symposium on System Identification*, 2003, pp. 331–336.

[15] I. Arasaratnam and S. Haykin, "Cubature Kalman Filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[16] S. Särkkä, "Unscented Rauch-Tung-Striebel Smoother," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, 2008.

[17] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo Smoothing for Nonlinear Time Series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 438–449, 2004.

[18] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[19] M. P. Deisenroth and H. Ohlsson, "A General Perspective on Gaussian Filtering and Smoothing: Explaining Current and Deriving New Algorithms," in *American Control Conference*, 2011.

[20] D. J. C. MacKay, "Introduction to Gaussian Processes," in *Neural Networks and Machine Learning*. Springer, 1998, vol. 168, pp. 133–165.

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

[22] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Local Gaussian Process Regression for Real Time Online Model Learning," in *Advances in Neural Information Processing Systems*, 2009, pp. 1193–1200.

[23] R. Murray-Smith, D. Sbarbaro, C. E. Rasmussen, and A. Girard, "Adaptive, Cautious, Predictive Control with Gaussian Process Priors," in *Symposium on System Identification*, 2003.

[24] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and B. Likar, "Predictive Control with Gaussian Process Models," in *IEEE Region 8 Eurocon 2003: Computer as a Tool*, 2003, pp. 352–356.

[25] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning," in *Robotics: Science & Systems*, 2011.

[26] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A Model-Based and

Data-Efficient Approach to Policy Search," in *International Conference on Machine Learning*, 2011.

[27] C. G. Atkeson and J. C. Santamaría, "A Comparison of Direct and Model-Based Reinforcement Learning," in *International Conference on Robotics and Automation*, 1997.

[28] J. Kern, "Bayesian Process-Convolution Approaches to Specifying Spatial Dependence Structure," Ph.D. dissertation, Institue of Statistics and Decision Sciences, Duke University, 2000.

[29] J. Quiñonero-Candela, A. Girard, J. Larsen, and C. E. Rasmussen, "Propagation of Uncertainty in Bayesian Kernel Models—Application to Multiple-Step Ahead Forecasting," in *International Conference on Acoustics, Speech and Signal Processing*, 2003, pp. 701–704.

[30] M. P. Deisenroth, *Efficient Reinforcement Learning using Gaussian Processes*. KIT Scientific Publishing, 2010, vol. 9.

[31] A. Doucet, S. J. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.