Gaussian process quadratures in nonlinear sigma-point filtering and smoothing

Gaussian Process Quadratures in Nonlinear Sigma-Point Filtering and Smoothing

Simo Särkkä*, Jouni Hartikainen†, Lennart Svensson‡ and Fredrik Sandblom§
*Aalto University, 02150 Espoo, Finland, simo.sarkka@aalto.fi
†Rocsole Ltd., 70150 Kuopio, Finland, jouni.hartikainen@rocsole.com
‡Chalmers University of Technology, SE-412 96 Gothenburg, Sweden, lennart.svensson@chalmers.se
§Volvo Group Trucks Technology, SE-405 08 Gothenburg, Sweden, fredrik.sandblom.2@volvo.com

***Abstract*—This paper is concerned with the use of Gaussian process regression based quadrature rules in the context of sigma-point-based nonlinear Kalman filtering and smoothing. We show how Gaussian process (i.e., Bayesian or Bayes–Hermite) quadratures can be used for numerical solving of the Gaussian integrals arising in the filters and smoothers. An interesting additional result is that with suitable selections of Hermite polynomial covariance functions the Gaussian process quadratures can be reduced to unscented transforms, spherical cubature rules, and to Gauss-Hermite rules previously proposed for approximate nonlinear Kalman filter and smoothing. Finally, the performance of the Gaussian process quadratures in this context is evaluated with numerical simulations.**

I. Introduction

The key computational challenge arising in many non-linear Kalman filtering and smoothing methods can be reduced to computation of Gaussian expectation integrals of the form

$$\mathcal{I}[\mathbf{g}] = \int \mathbf{g}(\mathbf{x})\ \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})\ \mathrm{d}\mathbf{x}, \tag{1}$$

where $\mathbf{g} : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a non-linear function. The reason to this is that the equations of non-linear Gaussian (Kalman) filters and smoothers [1]–[5] consist of Gaussian integrals of the above form and linear operations on vectors and matrices.

For example, the multidimensional Gaussian type of Gauss–Hermite quadrature and cubature[1] based filters and smoothers [3], [4], [6]–[8] are based on explicit numerical integration of the Gaussian integrals. The unscented transform based methods as well as other sigma-point methods [9]–[17] can also be retrospectively interpreted to belong to the class of Gaussian numerical integration based methods [6]. Conversely, Gaussian type of quadrature or cubature based methods can also be interpreted to be special cases of sigma-point methods. Furthermore, the classical Taylor series based methods [18] and Stirling's interpolation based methods [17], [19] can be seen as ways to approximate the integrand such that the Gaussian integral becomes tractable (cf. [5]). The recent Fourier–Hermite series [20], Hermite polynomial [21] methods are also based on numerical approximation of the integrands.

In this paper, we study the use of more recent numerical integration methods called Gaussian process quadratures [22]–[26] in the filtering and smoothing context. This kind of quadrature methods is also often referred to as Bayesian or Bayes–Hermite quadratures. They are closely related to Gauss–Hermite type of Gaussian quadratures in the sense that as Gaussian quadratures can be seen to form a polynomial approximation to the integrand via point-evaluations, Gaussian process quadratures use a Gaussian process regression [27] approximation instead [24]. Because Gaussian process regressors can be used to approximate much larger class of functions than polynomial approximations, they can be expected to perform much better in numerical integration than polynomial approximation based methods.

The use of Gaussian process regression approximations in the context of non-linear Kalman filtering and smoothing has been recently studied in [28], where the idea was to form a fixed Gaussian process approximation to the non-linearities allowing for closed form integration of the Gaussian integrals in the filtering and smoothing equations. Here we take a bit different, the Gaussian process quadrature point of view. That is, instead of training the Gaussian process beforehand from a set of point-wise evaluations of the dynamic function as in [28], we retrain the Gaussian process at every step to give a local approximation to the function. The resulting method is equivalent to the use of Gaussian process quadrature for approximating the Gaussian integrals in filtering and smoothing equations. This point of view has also been discussed [29] and [5]; and the aim of this article is to present these ideas in full as well as to analyze connections to sigma-point methods. In particular, we show the explicit relationship of the proposed methods to the unscented transform, cubature integration, and Gauss–Hermite quadrature based methods.

II. Background

A. Non-Linear Gaussian (Kalman) Filtering and Smoothing

In [3]–[6] sigma-point filtering and smoothing methods are treated as special cases of non-linear Gaussian (Kalman) filters and smoothers. These methods can be used to approximate the filtering distributions $p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_k)$ and smoothing distributions $p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$ of non-linear state-space models of the form

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}, \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k, \end{aligned} \tag{2}$$

where, for $k = 1, 2, \ldots, T$, $\mathbf{x}_k \in \mathbb{R}^n$ are the hidden states, $\mathbf{y}_k \in \mathbb{R}^d$ are the measurements, and $\mathbf{q}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ and $\mathbf{r}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_k)$ are the process and measurements noises, respectively. The non-linear function $\mathbf{f}(\cdot)$ is used to model the

[1] Terms *quadrature* and *cubature* simply mean numerical integration, the latter usually over more than a single variable.

dynamics of the system and $\mathbf{h}(\cdot)$ models the mapping from the states to the measurements.

Non-linear Gaussian filters (see, e.g., [5], page 98) are general methods to produce Gaussian approximations to the filtering distrubutions:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_k) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k), \quad k = 1, 2, \ldots, T. \tag{3}$$

Non-linear Gaussian smoothers (see, e.g., [5], page 154) are the corresponding methods to produce approximations to the smoothing distributions:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k^s, \mathbf{P}_k^s), \quad k = 1, 2, \ldots, T. \tag{4}$$

Both Gaussian filters and smoothers above can be easily generalized to state-space models with non-additive noises (see [5]), but here we only consider the additive noise case.

B. Gaussian Integration and Sigma-Point Methods

Sigma-point filtering and smoothing methods can be quite generally seen as methods which approximate the Gaussian integrals in the Gaussian filtering and smoothing equations as

$$\int \mathbf{g}(\mathbf{x}) \; \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) \; \mathrm{d}\mathbf{x} \approx \sum_i W_i \, \mathbf{g}(\mathbf{x}_i), \tag{5}$$

where W_i are some predefined weights and $\mathbf{x}_i$ are the sigma-points (classically called abscissas). Typically, the sigma-point methods use so called *stochastic decoupling* which refers to the idea that we do a change of variables

$$\int \mathbf{g}(\mathbf{x}) \; \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) \; \mathrm{d}\mathbf{x} = \int \underbrace{\mathbf{g}(\mathbf{m} + \sqrt{\mathbf{P}}\,\boldsymbol{\xi})}_{\tilde{\mathbf{g}}(\boldsymbol{\xi})} \; \mathrm{N}(\boldsymbol{\xi} \mid 0, \mathbf{I}) \; \mathrm{d}\boldsymbol{\xi} \tag{6}$$

where $\mathbf{P} = \sqrt{\mathbf{P}}\,\sqrt{\mathbf{P}}^{\mathsf{T}}$. This implies that we only need to design weights W_i and unit sigma-points $\boldsymbol{\xi}_i$ for integrating against unit Gaussian distributions:

$$\int \tilde{\mathbf{g}}(\boldsymbol{\xi}) \; \mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0}, \mathbf{I}) \; \mathrm{d}\boldsymbol{\xi} \approx \sum_i W_i \, \tilde{\mathbf{g}}(\boldsymbol{\xi}_i), \tag{7}$$

thus leading to approximations of the form

$$\int \mathbf{g}(\mathbf{x}) \; \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) \; \mathrm{d}\mathbf{x} \approx \sum_i W_i \, \mathbf{g}(\mathbf{m} + \sqrt{\mathbf{P}}\,\boldsymbol{\xi}_i). \tag{8}$$

Different sigma-point methods correspond to different choices of weights W_i and unit sigma-points $\boldsymbol{\xi}_i$. For example, the canonical unscented transform [9] uses the following set of $2n + 1$ weights and sigma-points:

$$\begin{aligned} W_0 &= \frac{\kappa}{n + \kappa}, \quad W_i = \frac{1}{2(n + \kappa)}, \quad i = 1, \ldots, 2n, \\ \boldsymbol{\xi}_i &= \begin{cases} \mathbf{0}, & i = 0, \\ \sqrt{n + \kappa}\,\mathbf{e}_i, & i = 1, \ldots, n, \\ -\sqrt{n + \kappa}\,\mathbf{e}_{i-n}, & i = n + 1, \ldots, 2n. \end{cases} \end{aligned} \tag{9}$$

where κ is a parameter of the algorithm and $\mathbf{e}_i$ is the unit vector towards the direction of the ith coordinate axis.

Note that sigma-point methods sometimes use different weights for the integrals appearing in the mean and covariance computations of Gaussian filters and smoothers. However, here we will only concentrate on the methods which use the same weights for both in order to derive more direct connections between the methods. For example, the above unscented transform weights are just a special case of more general unscented transforms (see, e.g., [5]).

C. Gaussian Process Regression

Gaussian process quadrature [23], [24] is based on forming a Gaussian process (GP) regression [27] approximation to the integrand using pointwise evaluations and then integrating the approximation. GP regression [27] is considered with predicting the value of an unknown function

$$o = g(\mathbf{x}) \tag{10}$$

at a certain test point $(o^*, \mathbf{x}^*)$ based on a finite number of training samples $\mathcal{D} = \{(o_j, \mathbf{x}_j) : j = 1, \ldots, N\}$ observed from it. The difference to classical regression is that instead of postulating a parametric regression function $g_\theta(\mathbf{x}; \boldsymbol{\theta})$, there $\boldsymbol{\theta} \in \mathbb{R}^D$ are the parameters, in GP regression we put a Gaussian process prior with a given covariance function $K(\mathbf{x}, \mathbf{x}')$ on the unknown functions $g_K(\mathbf{x})$.

In practice, the observations are often assumed to contain noise and hence a typical model setting is:

$$\begin{aligned} g_K &\sim \mathrm{GP}(0, K(\mathbf{x}, \mathbf{x}')) \\ o_j &= g_K(\mathbf{x}_j) + \epsilon_j, \quad \epsilon_j \sim \mathrm{N}(0, \sigma^2), \end{aligned} \tag{11}$$

where the first line above means that the random function g_K has a zero mean Gaussian process prior with the given covariance function $K(\mathbf{x}, \mathbf{x}')$. A commonly used covariance function is the exponentiated quadratic (also called squared exponential) covariance function

$$K(\mathbf{x}, \mathbf{x}') = s^2 \, \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|^2\right), \tag{12}$$

where $s, \ell > 0$ are parameters of the covariance function (see [27]). The GP regression equations can be derived as follows. Assume that we want to estimate the value of the "clean" function $g(\mathbf{x}^*)$ based on its Gaussian process approximation $g_K(\mathbf{x}^*)$ at a test point $\mathbf{x}^*$ given the vector of observed values $\mathbf{o} = (o_1, \ldots, o_N)$. Due to the Gaussian process assumption we now get

$$\begin{pmatrix} \mathbf{o} \\ g_K(\mathbf{x}^*) \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{K} + \sigma^2\mathbf{I} & \mathbf{k}(\mathbf{x}^*) \\ \mathbf{k}^{\mathsf{T}}(\mathbf{x}^*) & K(\mathbf{x}^*, \mathbf{x}^*) \end{pmatrix}\right) \tag{13}$$

where $\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)]$ is the joint covariance of observed points, $K(\mathbf{x}^*, \mathbf{x}^*)$ is the (co)variance of the test point, $\mathbf{k}(\mathbf{x}^*) = [K(\mathbf{x}^*, \mathbf{x}_i)]$ is the vector cross covariances with the test point.

The Bayesian estimate of the unknown value of $g_K(\mathbf{x}^*)$ is now given by its posterior distribution, given the training data. Because everything is Gaussian, the posterior distribution is Gaussian and hence by the posterior mean and variance:

$$\begin{aligned} \mathrm{E}[g_K(\mathbf{x}^*) \,|\, \mathbf{o}] &= \mathbf{k}^{\mathsf{T}}(\mathbf{x}^*) \, (\mathbf{K} + \sigma^2\mathbf{I})^{-1} \, \mathbf{o} \\ \mathrm{Var}[g_K(\mathbf{x}^*) \,|\, \mathbf{o}] &= K(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^{\mathsf{T}}(\mathbf{x}^*) \, (\mathbf{K} + \sigma^2\mathbf{I})^{-1} \mathbf{k}(\mathbf{x}^*). \end{aligned} \tag{14}$$

These are the Gaussian process regression equations in their typical form [27], in the special case where g is scalar.

The extension to multiple output dimensions is conceptually straightforward (see, e.g., [27], [30]), but construction of the covariance functions as well as the practical computational methods tend to be complicated [31], [32]. However, a typical easy approach to the multivariate case is to treat each of the dimensions independently.

D. Gaussian Process Quadrature

In Gaussian process quadrature [23], [24] the basic idea is to approximate the integral of a given function g (assumed scalar for simplicity) against a weight function $w(\mathbf{x})$ by evaluating the function g at a finite number of points and then by forming a Gaussian process approximation g_K to the function. The integral is then approximated by integrating the Gaussian process approximation (or its posterior mean) which is conditioned on the evaluation points instead of the function itself. That is, we use

$$\begin{aligned}\int g(\mathbf{x})\, w(\mathbf{x})\, \mathrm{d}\mathbf{x} &\approx \int \mathrm{E}[g_K(\mathbf{x}) \,|\, \mathbf{o}]\, w(\mathbf{x})\, \mathrm{d}\mathbf{x} \\ &= \left[\int \mathbf{k}^{\mathsf{T}}(\mathbf{x})\; w(\mathbf{x})\; \mathrm{d}\mathbf{x}\right] (\mathbf{K} + \sigma^2 \mathbf{I})^{-1}\, \mathbf{o},\end{aligned} \tag{15}$$

where the "training set" $\mathbf{o} = (g(\mathbf{x}_1), \ldots, g(\mathbf{x}_N))$ now contains the values of the function g evaluated at certain selected inputs. The posterior variance of the integral can be evaluated in an analogous manner, and it is sometimes used to optimize the evaluation points of the function g_N [23]–[26].

If in (15) we denote the ith component of the row matrix $\left[\int \mathbf{k}^{\mathsf{T}}(\mathbf{x})\; w(\mathbf{x})\; \mathrm{d}\mathbf{x}\right]\; (\mathbf{K}+\sigma^2\mathbf{I})^{-1}$ as W_i the Gaussian process quadrature approximation can be seen to have the form

$$\int g(\mathbf{x})\, w(\mathbf{x})\; \mathrm{d}\mathbf{x} \approx \sum_i W_i\, g(\mathbf{x}_i), \tag{16}$$

which is both a Gaussian quadrature and a sigma-point type of approximation.

As in the classical Gaussian quadrature integration the selection of evaluation points $\mathbf{x}_i$ affects the quality of the integral approximation considerably. The classical quadrature points are determined by the roots of special polynomials (e.g., Hermite polynomials). In the Gaussian process quadratures we can either use some predefined point designs (such as the unscented transform or Gauss-Hermite points [5]), or optimize the point set using a suitable criterion [23]–[26].

III. Gaussian Process Quadratures in Non-Linear Filtering and Smoothing

A. Gaussian Process Quadrature Based Sigma-Point Methods

The key innovation, which makes Gaussian quadrature integration useful in the present sigma-point filtering and smoothing context, is that when the weight function is Gaussian $w(\mathbf{x}) = \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})$, we can often evaluate the term $\int \mathbf{k}(\mathbf{x})\, \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})\; \mathrm{d}\mathbf{x}$ in closed form. This happens, for example, when the covariance function is the exponentiated quadratic (12), which leads to the classical Bayes–Hermite integration [23], and it is also closely related to the Gaussian process regression based filters proposed in [28].

If we use stochastic decoupling (6) with the Gaussian process quadrature, it turns out that the weights W_i and unit sigma-points can be fully precomputed $\boldsymbol{\xi}_i$, because they are independent of the means and covariances. Hence we obtain the following algorithm which uses Gaussian process quadrature to approximate multidimensional Gaussian integration required in the Gaussian filters and smoothers.

Algorithm III.1 (Gaussian process quadrature based sigma–point approximation)**.** *The Gaussian process quadrature (or Bayes–Hermite/Bayesian quadrature) based sigma-point-type of integral approximation can be computed as*

$$\int \mathbf{g}(\mathbf{x})\; \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})\; \mathrm{d}\mathbf{x} \approx \sum_{i=1}^{N} W_i\, \mathbf{g}(\mathbf{x}_i), \tag{17}$$

where $\mathbf{x}_i = \mathbf{m} + \sqrt{\mathbf{P}}\, \boldsymbol{\xi}_i$ with the unit sigma-points $\boldsymbol{\xi}_i$ are selected according to a predefined criterion (see discussion in Section II-D), and the weights are determined by

$$W_i = \left[\int \mathbf{k}^{\mathsf{T}}(\boldsymbol{\xi})\, \mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0}, \mathbf{I})\; \mathrm{d}\boldsymbol{\xi}\, (\mathbf{K} + \sigma^2 \mathbf{I})^{-1}\right]_i, \tag{18}$$

where $\mathbf{K} = [K(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)]$ is the matrix of unit sigma-point covariances and $\mathbf{k}(\boldsymbol{\xi}) = [K(\boldsymbol{\xi}, \boldsymbol{\xi}_i)]$ is the vector cross covariances. In principle, the choice of sigma-points above is completely free, but good choices of them are discussed below.

The weights W_i in Algorithm III.1 are fully determined once the covariance function and the unit sigma-points $\boldsymbol{\xi}_i$ are fixed. The above algorithm implicitly models the components of $\mathbf{g}(\mathbf{x})$ as independent Gaussian processes with identical covariance, but generalization to different covariance functions and dependent processes is possible as well (cf. [30]–[32]). Note that, for example, with the exponentiated quadratic covariance function (12), the integral $\int \mathbf{k}^{\mathsf{T}}(\boldsymbol{\xi})\, \mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0}, \mathbf{I})\; \mathrm{d}\boldsymbol{\xi}$ can be computed easily in closed form (see [28]) and hence the weights are available in closed form as well.

The selection of unit sigma-points $\boldsymbol{\xi}_i$ is an important factor in the accuracy of the above sigma-point approximation – along with choice of covariance function, of course. Good selections are, for example:

1) The sigma-points of the unscented transform (9).
2) Quasi-random point patterns such as Hammersley points sets [33], which are transformed through the inverse of a cumulative Gaussian distribution.
3) Multivariate Gauss–Hermite abscissas [3], [5].
4) Minimum variance point sets [23], [24].

In the theoretical analysis of this paper, we concentrate on the unscented transform points, because they provide an explicit link to GP quadrature based sigma-point methods. In the numerical experiments we also use Hammersley point sets, because they turned out to perform well in practice and are easy to compute.

B. Gaussian Process Transform

We can also define a general Gaussian process transform as follows (cf. [5]).

Algorithm III.2 (Gaussian process transform)**.** *The Gaussian process quadrature based Gaussian approximation to the joint*

distribution of $\mathbf{x}$ *and the transformed random variable* $\mathbf{y} = \mathbf{g}(\mathbf{x}) + \mathbf{q}$*, where* $\mathbf{x} \sim \mathrm{N}(\mathbf{m}, \mathbf{P})$ *and* $\mathbf{q} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q})$*, is given by*

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mathbf{m} \\ \boldsymbol{\mu}_{\mathrm{GP}} \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{C}_{\mathrm{GP}} \\ \mathbf{C}_{\mathrm{GP}}^{\mathsf{T}} & \mathbf{S}_{\mathrm{GP}} \end{pmatrix} \right), \tag{19}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i \, \mathbf{g}(\mathbf{x}_i), \\ \mathbf{S}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i \, (\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}}) \, (\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}})^{\mathsf{T}} + \mathbf{Q}, \\ \mathbf{C}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i \, (\mathbf{x}_i - \mathbf{m}) \, (\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}})^{\mathsf{T}}, \end{aligned} \tag{20}$$

where the weights W_i *and sigma-points* $\mathbf{x}_i$ *are selected according to Algorithm III.1.*

C. Gaussian Process Quadrature Filter and Smoother

In this section we show how to construct filters and smoothers using the Gaussian process quadrature approximations. Because Algorithm III.1 can be seen as a sigma-point method, analogously to other sigma-point filters considered, for example, in [5], we can now formulate the following sigma-point filter for model (2), which uses the unit sigma-points $\boldsymbol{\xi}_i$ and weights W_i defined by Algorithm III.1.

Algorithm III.3 (Gaussian process quadrature filter)**.** *The filtering is started from initial mean and covariance,* $\mathbf{m}_0$ *and* $\mathbf{P}_0$*, respectively, such that* $\mathbf{x}_0 \sim \mathrm{N}(\mathbf{m}_0, \mathbf{P}_0)$*. Then the following prediction and update steps are applied for* $k = 1, 2, 3, \ldots, T$*.*

Prediction:

1) *Form the sigma points as follows:* $\mathcal{X}_{k-1}^{(i)} = \mathbf{m}_{k-1} + \sqrt{\mathbf{P}_{k-1}}\, \boldsymbol{\xi}_i, i = 1, \ldots, N$*.*
2) *Propagate the sigma points through the dynamic model:* $\hat{\mathcal{X}}_k^{(i)} = \mathbf{f}(\mathcal{X}_{k-1}^{(i)}), i = 1, \ldots, N$*.*
3) *Compute the predicted mean* $\mathbf{m}_k^-$ *and the predicted covariance* $\mathbf{P}_k^-$*:*

$$\begin{aligned} \mathbf{m}_k^- &= \sum_{i=1}^{N} W_i \, \hat{\mathcal{X}}_k^{(i)}, \\ \mathbf{P}_k^- &= \sum_{i=1}^{N} W_i \, (\hat{\mathcal{X}}_k^{(i)} - \mathbf{m}_k^-) \, (\hat{\mathcal{X}}_k^{(i)} - \mathbf{m}_k^-)^{\mathsf{T}} + \mathbf{Q}_{k-1}. \end{aligned}$$

Update:

1) *Form the sigma points:* $\mathcal{X}_k^{-(i)} = \mathbf{m}_k^- + \sqrt{\mathbf{P}_k^-}\, \boldsymbol{\xi}_i, i = 1, \ldots, N$*.*
2) *Propagate sigma points through the measurement model:* $\hat{\mathcal{Y}}_k^{(i)} = \mathbf{h}(\mathcal{X}_k^{-(i)}), i = 1 \ldots N$*.*
3) *Compute the predicted mean* $\boldsymbol{\mu}_k$*, the predicted covariance of the measurement* $\mathbf{S}_k$*, and the cross-covariance of the state and the measurement* $\mathbf{C}_k$*:*

$$\begin{aligned} \boldsymbol{\mu}_k &= \sum_{i=1}^{N} W_i \, \hat{\mathcal{Y}}_k^{(i)}, \\ \mathbf{S}_k &= \sum_{i=1}^{N} W_i \, (\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k) \, (\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k)^{\mathsf{T}} + \mathbf{R}_k, \\ \mathbf{C}_k &= \sum_{i=1}^{N} W_i \, (\mathcal{X}_k^{-(i)} - \mathbf{m}_k^-) \, (\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k)^{\mathsf{T}}. \end{aligned}$$

4) *Compute the filter gain* $\mathbf{K}_k$ *and the filtered state mean* $\mathbf{m}_k$ *and covariance* $\mathbf{P}_k$*, conditional on the measurement* $\mathbf{y}_k$*:*

$$\begin{aligned} \mathbf{K}_k &= \mathbf{C}_k \, \mathbf{S}_k^{-1}, \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k \, [\mathbf{y}_k - \boldsymbol{\mu}_k], \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \, \mathbf{S}_k \, \mathbf{K}_k^{\mathsf{T}}. \end{aligned}$$

Further following the line of thought in [5] we can formulate a sigma-point smoother using the unit sigma-points and weights from Algorithm III.1.

Algorithm III.4 (Gaussian process quadrature sigma-point RTS smoother)**.** *The smoothing recursion is started from the filtering result of the last time step* $k = T$*, that is,* $\mathbf{m}_T^s = \mathbf{m}_T$*,* $\mathbf{P}_T^s = \mathbf{P}_T$ *and proceeded backwards for* $k = T - 1, T - 2, \ldots, 1$ *as follows.*

1) *Form the sigma points:* $\mathcal{X}_k^{(i)} = \mathbf{m}_k + \sqrt{\mathbf{P}_k}\, \boldsymbol{\xi}_i, i = 1, \ldots, N$*.*
2) *Propagate the sigma points through the dynamic model:* $\hat{\mathcal{X}}_{k+1}^{(i)} = \mathbf{f}(\mathcal{X}_k^{(i)}), i = 1, \ldots, N$*.*
3) *Compute the predicted mean* $\mathbf{m}_{k+1}^-$*, the predicted covariance* $\mathbf{P}_{k+1}^-$*, and the cross-covariance* $\mathbf{D}_{k+1}$*:*

$$\begin{aligned} \mathbf{m}_{k+1}^- &= \sum_{i=1}^{N} W_i \, \hat{\mathcal{X}}_{k+1}^{(i)}, \\ \mathbf{P}_{k+1}^- &= \sum_{i=1}^{N} W_i \, (\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-) \, (\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-)^{\mathsf{T}} + \mathbf{Q}_k, \\ \mathbf{D}_{k+1} &= \sum_{i=1}^{N} W_i \, (\mathcal{X}_k^{(i)} - \mathbf{m}_k) \, (\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-)^{\mathsf{T}}. \end{aligned}$$

4) *Compute the gain* $\mathbf{G}_k$*, mean* $\mathbf{m}_k^s$ *and covariance* $\mathbf{P}_k^s$ *as follows:*

$$\begin{aligned} \mathbf{G}_k &= \mathbf{D}_{k+1} \, [\mathbf{P}_{k+1}^-]^{-1}, \\ \mathbf{m}_k^s &= \mathbf{m}_k + \mathbf{G}_k \, (\mathbf{m}_{k+1}^s - \mathbf{m}_{k+1}^-), \\ \mathbf{P}_k^s &= \mathbf{P}_k + \mathbf{G}_k \, (\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1}^-) \, \mathbf{G}_k^{\mathsf{T}}. \end{aligned}$$

Note that we could cope with non-additive noises in the model by using augmented forms of the above filters and smoothers as in [5].

IV. Relationship with Other Sigma-Point Methods

In this section we start by briefly reviewing multivariate Fourier–Hermite series and discuss its relationship with

multivariate Gaussian integration. We then show how the unscented transform can be seen as a method to determine the coefficients of a truncated Fourier–Hermite series from pointwise-evaluations of the function. Finally, we construct a covariance function which makes the Gaussian process transform equivalent to the unscented transform. Because we can always transform integration over a general Gaussian distribution $\mathrm{N}(\mathbf{m}, \mathbf{P})$ into integration over a unit Gaussian $\mathrm{N}(\mathbf{0}, \mathbf{I})$ via the use of stochastic decoupling (6), we will restrict the analysis here to unit Gaussian distributions.

A. Fourier–Hermite Series

Fourier–Hermite series (see, e.g., [34]) are orthogonal polynomial series in a Hilbert space, where the inner product is defined via an expectation of a product over a Gaussian distributions. These series are also inherently related to non-linear Gaussian filtering as they can be seen as generalizations of statistical linearization and they also have a deep connection with unscented transforms, Gaussian quadrature integration, and Gaussian process regression [5], [20], [21].

We can define an inner product of multivariate scalar functions $f(\mathbf{x})$ and $g(\mathbf{x})$ as follows:

$$\langle f, g \rangle = \int f(\mathbf{x})\, g(\mathbf{x})\, \mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I})\, \mathrm{d}\mathbf{x}. \tag{21}$$

If we now define a norm via $||f||_{\mathcal{H}}^2 = \langle f, f \rangle$, and the corresponding distance function $d(f, g) = ||f - g||_{\mathcal{H}}$, then the functions $||f||_{\mathcal{H}} < \infty$ form a Hilbert space $\mathcal{H}$. It now turns out that the multivariate Hermite polynomials form a complete orthogonal basis of the resulting Hilbert space [34].

A multivariate Hermite polynomial with multi-index $\mathcal{I} = \{i_1, \ldots, i_n\}$ can be defined as

$$H_{\mathcal{I}}(\mathbf{x}) = H_{i_1}(x_1) \times \cdots \times H_{i_n}(x_n) \tag{22}$$

which is a product of univariate Hermite polynomials

$$H_p(x) = (-1)^p \exp(x^2/2) \frac{\mathrm{d}^p}{\mathrm{d}x^p} \exp(-x^2/2). \tag{23}$$

The orthogonality property can now be expressed as

$$\langle H_{\mathcal{I}}, H_{\mathcal{J}} \rangle = \begin{cases} \mathcal{I}!, & \text{if } \mathcal{I} = \mathcal{J} \\ 0, & \text{otherwise,} \end{cases} \tag{24}$$

where we have denoted $\mathcal{I}! = i_1! \cdots i_n!$ and $\mathcal{I} = \mathcal{J}$ means that each of the elements in the multi-indices $\mathcal{I} = \{i_1, \ldots, i_n\}$ and $\mathcal{J} = \{j_1, \ldots, j_n\}$ are equal. We will also denote the sum of indices as $|\mathcal{I}| = i_1 + \cdots + i_n$.

A function $g(\mathbf{x})$ with $\langle g, g \rangle < \infty$ can be expanded into Fourier–Hermite series [34]

$$g(\mathbf{x}) = \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} c_{\mathcal{I}}\, H_{\mathcal{I}}(\mathbf{x}), \tag{25}$$

where $H_{\mathcal{I}}(\mathbf{x})$ are multivariate Hermite polynomials and the series coefficients are given by the inner products $c_{\mathcal{I}} = \langle H_{\mathcal{I}}, g \rangle$.

The Fourier–Hermite series representation is particularly useful if we are interested in computing the integral of the function over a unit Gaussian distribution. From the definition of Fourier–Hermite series it follows that the expected value of the function is simply

$$\int g(\mathbf{x})\, \mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I})\, \mathrm{d}\mathbf{x} = \langle H_{\mathbf{0}}, g \rangle = c_{\mathbf{0}}. \tag{26}$$

Expectations of vector valued functions can be similarly extracted from the zeroth order terms of the component-wise Fourier–Hermite series. The covariance can then also be computed by a Parseval-like identity given in [20]. Thus, if we knew the coefficients of the Fourier–Hermite series, the construction of the Gaussian moment matching transform would be possible without additional approximations, because all the required expectations can be extracted from the series coefficients. However, in practice, we rarely know the coefficients of the series nor can we easily compute them for a given non-linear function.

B. Relationship of Fourier-Hermite Series and Unscented Transform

Consider a Gaussian process $g_G(\mathbf{x})$ which has zero mean and a covariance function $K(\mathbf{x}, \mathbf{x}')$. In the same way as deterministic functions, Gaussian processes can also be expanded into Fourier–Hermite series:

$$g_G(\mathbf{x}) = \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} \tilde{c}_{\mathcal{I}}\, H_{\mathcal{I}}(\mathbf{x}), \tag{27}$$

where the coefficients are given as $\tilde{c}_{\mathcal{I}} = \langle H_{\mathcal{I}}, g_G \rangle$. The coefficients $\tilde{c}_{\mathcal{I}}$ are zero mean Gaussian random variables and their covariance is given as

$$\begin{aligned} \mathrm{E}\left[\tilde{c}_{\mathcal{I}}\, \tilde{c}_{\mathcal{J}}\right] &= \mathrm{E}\left[\langle H_{\mathcal{I}}, g_G \rangle \langle H_{\mathcal{J}}, g_G \rangle\right] \\ &= \iint H_{\mathcal{I}}(\mathbf{x})\, K(\mathbf{x}, \mathbf{x}')\, H_{\mathcal{J}}(\mathbf{x}') \\ &\qquad \times \mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I})\, \mathrm{N}(\mathbf{x}' \mid \mathbf{0}, \mathbf{I})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{x}'. \end{aligned} \tag{28}$$

If we define constants $\lambda_{\mathcal{I},\mathcal{J}} = \mathrm{E}\left[\tilde{c}_{\mathcal{I}}\, \tilde{c}_{\mathcal{J}}\right]$ then the covariance function $K(\mathbf{x}, \mathbf{x}')$ can be further written as series

$$K(\mathbf{x}, \mathbf{x}') = \sum_{q=0}^{\infty} \sum_{|\mathcal{J}|=q} \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!\, \mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}}\, H_{\mathcal{I}}(\mathbf{x})\, H_{\mathcal{J}}(\mathbf{x}'). \tag{29}$$

Recently, Sandblom and Svensson [21] proposed the marginalized transform where the idea is to identify the coefficients in a Hermite polynomial series (i.e., Fourier–Hermite series) by point-wise evaluations of the function $g(\mathbf{x})$. When the series coefficients are treated as unknown parameters in Bayesian sense, as was done in [21], this approach can be interpreted to form a Gaussian process approximation to the unknown function.

Recall that the Bayesian way to treating the unknown series coefficients $c_{\mathcal{I}}$ in the series (25) is by replacing them with random variables $\tilde{c}_{\mathcal{I}}$ and computing their posterior distribution given the point-wise evaluations (observations). The result of replacing the coefficients with Gaussian random variables results exactly in a series expansion of the form (27). In principle, it would now be possible to compute the posterior distribution of the whole series of coefficients given the finite set of observations. However, to make computations tractable, it is convenient to truncate the series.

Assume now that the coefficients of order four and above $|\mathcal{I}| \geq 4$ vanish and thus the function is in fact a third order multivariate polynomial. The corresponding prior Gaussian random function is then given as

$$g_G(\mathbf{x}) = \sum_{p=0}^{3} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} \tilde{c}_{\mathcal{I}} \, H_{\mathcal{I}}(\mathbf{x}). \tag{30}$$

Let us evaluate the integrand g in the unscented transform points $\mathbf{x}_i \triangleq \boldsymbol{\xi}_i$ given in (9). If we let $z_i = g(\mathbf{x}_i)$ for $i = 0, \ldots, 2n$, and denote $\mathbf{z} = \{z_i \ : \ i = 0, \ldots, 2n\}$, $\tilde{\mathbf{c}} = \{\tilde{c}_{\mathcal{I}} \ : \ |\mathcal{I}| \leq p\}$ then the solution to the estimation problem on the coefficients is the posterior distribution

$$p(\tilde{\mathbf{c}} \mid \mathbf{z}) = \frac{p(\mathbf{z} \mid \tilde{\mathbf{c}}) \, p(\tilde{\mathbf{c}})}{\int p(\mathbf{z} \mid \tilde{\mathbf{c}}) \, p(\tilde{\mathbf{c}}) \, \mathrm{d}\tilde{\mathbf{c}}} \tag{31}$$

Recall that we are now actually interested in computing the integral of $g(\mathbf{x})$ over the Gaussian distribution $\mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I})$. Due to Equation (26) we are interested in the marginal posterior distribution of the zeroth order coefficient:

$$p(\tilde{c}_{\mathbf{0}} \mid \mathbf{z}) = \int p(\tilde{\mathbf{c}} \mid \mathbf{z}) \, \mathrm{d}\tilde{\mathbf{c}}_{\mathcal{I} \neq \mathbf{0}}. \tag{32}$$

It now turns out that the above distribution is actually singular as stated by the following theorem which is a generalization of the result derived in [21].

Theorem IV.1 (Posterior of $\tilde{c}_{\mathbf{0}}$)**.** *The posterior distribution of the zeroth order coefficient $\tilde{c}_{\mathbf{0}}$ is*

$$p(\tilde{c}_{\mathbf{0}} \mid \mathbf{z}) = \delta(\tilde{c}_{\mathbf{0}} - c_{\mathbf{0}}), \tag{33}$$

where $\delta(\cdot)$ is the Dirac delta distribution and $c_{\mathbf{0}}$ is given as

$$c_{\mathbf{0}} = W_0 \, z_i + W \sum_{i=1}^{2n} z_i, \quad W_0 = \frac{\kappa}{n+\kappa}, \quad W = \frac{1}{2(n+\kappa)}. \tag{34}$$

Proof: It is sufficient to show that the following system of equations can be uniquely solved for $\tilde{c}_{\mathbf{0}}$ and the solution is $c_{\mathbf{0}}$ in Equation (34):

$$z_i = \sum_{p=0}^{3} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} \tilde{c}_{\mathcal{I}} \, H_{\mathcal{I}}(\mathbf{x}_i), \quad i = 0, \ldots, 2n. \tag{35}$$

First note that we have

$$z_i = \sum_{p=0}^{3} \sum_{|\mathcal{I}|=p} \begin{cases} \frac{\tilde{c}_{\mathcal{I}}}{\mathcal{I}!} \, H_{\mathcal{I}}(\mathbf{0}), & i = 0 \\ \frac{\tilde{c}_{\mathcal{I}}}{\mathcal{I}!} \, H_{\mathcal{I}}(\sqrt{n+\kappa} \, \mathbf{e}_i), & i = 1, \ldots, n \\ \frac{\tilde{c}_{\mathcal{I}}}{\mathcal{I}!} \, H_{\mathcal{I}}(-\sqrt{n+\kappa} \, \mathbf{e}_{i-n}), & i = n+1, \ldots, 2n. \end{cases} \tag{36}$$

Let us now attempt to simplify the expression of z_0. First note that $H_{\mathcal{I}}(\mathbf{0})$ is 1 when $\mathcal{I} = \mathbf{0}$, and zero whenever any of the elements of $\mathcal{I}$ is odd. The only purely even terms have univariate second order Hermite polynomials which give $H_2(0) = -1$. Thus we get

$$z_0 = \tilde{c}_{\mathbf{0}} - \frac{1}{2} \sum_{i=1}^{n} \tilde{c}_{2\mathbf{e}_i}, \tag{37}$$

where $2\mathbf{e}_i$ denotes a set of indices with 2 at index i and zeros elsewhere.

Let us now see what happens if we calculate the sum of terms $i = 1, \ldots, 2n$. Recall that the only zeroth order (multivariate) Hermite polynomial is $H_{\mathbf{0}}(\mathbf{x}) = 1$. Thus from the zeroth order terms we get $2n \, \tilde{c}_{\mathbf{0}}$. The first order terms are just univariate Hermite polynomials $H_1(x_i) = x_i$. Because we have $H_1(\sqrt{n+\kappa} \, x_i) + H_1(-\sqrt{n+\kappa} \, x_i) = 0$, all the first order terms simply cancel out. It is also easy to see that the third order terms are either zero or cancel out and we are only left with the second order terms. For each i we get two second order terms with coefficient $H_2(\pm\sqrt{n+\kappa}) = (n+\kappa) - 1$ and $2n - 2$ terms with $H_2(0) = -1$. Thus we get

$$\begin{aligned} \sum_{i=1}^{2n} z_i &= 2n \, \tilde{c}_{\mathbf{0}} + \sum_{i=1}^{n} \tilde{c}_{2\mathbf{e}_i} \left[(n+\kappa) - 1 \right] - \sum_{i=1}^{n} \sum_{j \neq i} \tilde{c}_{2\mathbf{e}_j} \\ &= 2n \, \tilde{c}_{\mathbf{0}} + \kappa \sum_{i=1}^{n} \tilde{c}_{2\mathbf{e}_i} \end{aligned} \tag{38}$$

and we are left with the pair of equations

$$\begin{aligned} z_0 &= \tilde{c}_{\mathbf{0}} - \frac{1}{2} \sum_{i=1}^{n} \tilde{c}_{2\mathbf{e}_i} \\ \sum_{i=1}^{2n} z_i &= 2n \, \tilde{c}_{\mathbf{0}} + \kappa \sum_{i=1}^{n} \tilde{c}_{2\mathbf{e}_i} \end{aligned} \tag{39}$$

Solving for $\tilde{c}_{\mathbf{0}}$ gives the unique solution

$$\tilde{c}_{\mathbf{0}} = \frac{\kappa}{n+\kappa} \, z_0 + \frac{1}{2(n+\kappa)} \sum_{i=1}^{2n} z_i, \tag{40}$$

which completes the proof. ∎

The above theorem thus states that if we do Bayesian inference on the Gaussian random function (30) using the values of function g evaluated at the unscented transform points, the posterior distribution of the expected value is concentrated at the unscented transform mean estimate.

C. Relationship of Gaussian Process Transform with Unscented Transform

We can now get an explicit link to Gaussian process regression by noting that the random function in Equation (30) is a Gaussian process with the covariance function

$$K(\mathbf{x}, \mathbf{x}') = \sum_{q=0}^{3} \sum_{|\mathcal{J}|=q} \sum_{p=0}^{3} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}! \, \mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} \, H_{\mathcal{I}}(\mathbf{x}) \, H_{\mathcal{J}}(\mathbf{x}'). \tag{41}$$

where the series coefficients are $\lambda_{\mathcal{I},\mathcal{J}} = \mathrm{E}\left[\tilde{c}_{\mathcal{I}} \, \tilde{c}_{\mathcal{J}}\right]$. The Gaussian process regression with the above covariance function now turns out to be equivalent to parameteric regression with the function (30) and hence we get the following theorem.

Theorem IV.2 (UT covariance function)**.** *If we select the GP covariance functions as in Equation* (41) *and use the UT points as the training set, then the GP transform is equivalent to the unscented transform [9].*

Proof: As shown in [27] computing predictions with the parametric model in Equation (30) and with a GP with covariance function (41) conditionally on a set of measurements are completely equivalent. This relationship is sometimes called

kernel trick. Thus prediction of the integral must be equivalent as well, because the expectation is just a linear functional of the predictor. ■

Corollary IV.1 (Spherical cubature covariance function)**.** *The Gaussian process transform reduces to the third cubature integration rule [7] when we select $\kappa = 0$ above and only use the evaluation points with indices $i = 1, \ldots, 2n$.*

D. Relationship with Gauss–Hermite and Other Methods

The cartesian product based Gauss–Hermite integration rule used in [3]–[6] is the unique integration method which is able to integrate all multivariate Hermite polynomials with $i_m \leq 2p-1$. The method can be interpreted to form a pth order multivariate polynomial product approximation to the integrand and selecting evaluation points in an optimal manner. Due to the uniqueness, if we take all Hermite polynomials with $i_m \leq p$ to a summation of the form (41) and use the Gauss–Hermite sigma-points, we recover the Gauss–Hermite method from the Gaussian process quadrature based method. Relationships with other methods could be derived from the results of [24] or by recalling that all parametric approximations can be considered as special cases of Gaussian process regression via the kernel trick [27].

V. Numerical Results

A. Covariance Function Implied by Unscented Transform

The exponentiated quadratic (i.e., the squared exponential) covariance function and the unscented transform covariance function (41) together with the corresponding Gaussian process regression results on random data are illustrated in Figure 1. The polynomial nature of the unscented transform covariance function can be clearly seen in the figures. The Gaussian process prediction with the unscented transform covariance function has a polynomial shape and thus it has less flexibility to explain the data than the exponentiated quadratic.

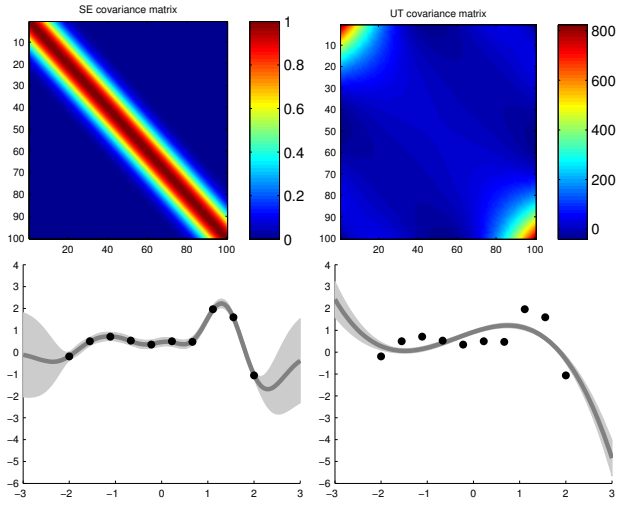

Fig. 1. *Top left:* Exponentiated quadratic (i.e., squared exponential covariance function. *Top right:* The covariance function implied by the unscented transform. *Bottom left:* Gaussian process regression with the exponentiated quadratic covariance function. *Bottom right:* Gaussian process regression with the covariance function implied by the unscented transform.

B. Univariate Non-linear Growth Model

In this section we compare the performance of the different methods in the following univariate non-linear growth model (UNGM) which is often used for benchmarking non-linear estimation methods:

$$\begin{aligned} x_k &= \frac{1}{2}\, x_{k-1} + 25\, \frac{x_{k-1}}{1 + x_{k-1}^2} + 8\, \cos(1.2\, k) + q_{k-1}, \\ y_k &= \frac{1}{20}\, x_k^2 + r_k, \end{aligned} \tag{42}$$

where $x_0 \sim \mathrm{N}(0, 5)$, $q_{k-1} \sim \mathrm{N}(0, 10)$, and $r_k \sim \mathrm{N}(0, 1)$.

We generated 100 independent datesets with 500 time steps each and applied the following methods to it: extended, unscented ($\kappa = 2$), and cubature filters and smoothers (EKF/UKF/CKF/ERTS/URTS/CRTS); Gauss–Hermite filters and smoothers with 3, 7, and 10 points (GHKF/GHRTS); Gaussian process quadrature filter and smoother with unscented transform points (GPQKFU/GPQRTSU); and with Hammersley point sets of sizes 3, 7, and 10 (GPQKFH/GPQRTSH). The covariance function was the exponentiated quadratic with $s = 1$ and $\ell = 3$ and the noise variance was set to 10^{-8}. The RMSE results (computed across the 500 time steps) together with single standard derivation bars are shown in Figures 2 and 3. As can be seen in the figures, with 5 and 10 points the Gaussian process quadrature based filters and smoothers have significantly lower errors than almost all the other methods – only Gauss–Hermite with 10 points and the cubature RTS smoother come close.

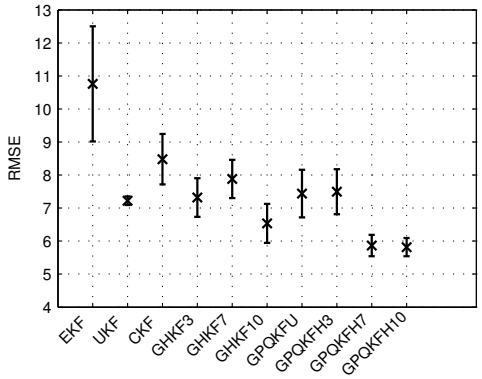

Fig. 2. RMSE results of filters in the UNGM experiment.

C. Pendulum Tracking

We also tested similar methods as in the previous section in the two-dimensional-state pendulum tracking example considered in [5]. For comparison, we also tested the Gaussian process filtering and smoothing methods (GPKF/GPRTS) proposed in [28] and taught a Gaussian process regression model from a training set of size 100. As already shown in [5], non-linear Kalman filters and smoothers tend to work practically identically in this particular problem. Our results indeed confirmed it such that 100 Monte Carlo runs did not reveal any statistically significant differences in the results. For this and space reason we also leave out the table of RMSE results here, because it would have identical RMSEs for all filters and similarly for the smoothers. Additional multidimensional examples will be considered in a future work.

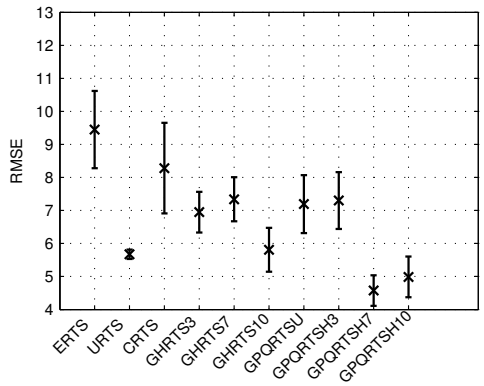

Fig. 3. RMSE results of smoothers in the UNGM experiment.

VI. Conclusion

In this paper we have shown how Gaussian process quadrature can be used as a sigma-point method for solving Gaussian integrals arising in non-linear Kalman filters and smoothers. We have also shown that many of the existing sigma-point methods can be derived as special cases of this method. Finally, via numerical simulations, we have shown that the proposed methodology also works well in practice.

Acknowledgment

The authors would like to thank Academy of Finland (projects 266940 and 273475) for funding.

References

[1] H. J. Kushner, "Approximations to optimal nonlinear filters," *IEEE Transactions on Automatic Control*, vol. 12, no. 5, pp. 546–556, October 1967.

[2] ——, "Numerical approximations to optimal nonlinear filters," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovskiĭ, Eds. Oxford, 2011, ch. 28.

[3] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45(5), pp. 910–927, 2000.

[4] S. Särkkä and J. Hartikainen, "On Gaussian optimal smoothing of non-linear state space models," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1938–1941, August 2010.

[5] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[6] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on Gaussian filters," *IEEE Transactions on Signal Processing*, vol. 54(8), pp. 2910–2921, 2006.

[7] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54(6), pp. 1254–1269, 2009.

[8] ——, "Cubature Kalman smoothers," *Automatica*, vol. 47, no. 10, pp. 2245–2250, 2011.

[9] S. J. Julier and J. K. Uhlmann, "A general method of approximating nonlinear transformations of probability distributions," Robotics Research Group, Department of Engineering Science, University of Oxford, Tech. Rep., 1995.

[10] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45(3), pp. 477–482, March 2000.

[11] E. A. Wan and R. Van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. Wiley, 2001, ch. 7.

[12] R. Van der Merwe, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, OGI School of Science & Engineering, Oregon Health & Science University, Portland, OR, USA, April 2004.

[13] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92(3), pp. 401–422, March 2004.

[14] S. Särkkä, "Recursive Bayesian inference on stochastic differential equations," Doctoral dissertation, Helsinki University of Technology, 2006.

[15] M. Šimandl and J. Duník, "Design of derivative-free smoothers and predictors," in *Preprints of the 14th IFAC Symposium on System Identification*, 2006, pp. 991–996.

[16] S. Särkkä, "Unscented Rauch-Tung-Striebel smoother," *IEEE Transactions on Automatic Control*, vol. 53(3), pp. 845–849, 2008.

[17] M. Šimandl and J. Duník, "Derivative-free estimation methods: New results and performance analysis," *Automatica*, vol. 45, no. 7, pp. 1749–1757, 2009.

[18] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[19] M. Nørgaard, N. K. Poulsen, and O. Ravn, "New developments in state estimation for nonlinear systems," *Automatica*, vol. 36(11), pp. 1627 – 1638, 2000.

[20] J. Sarmavuori and S. Särkkä, "Fourier-Hermite Kalman filter," *IEEE Transactions on Automatic Control*, vol. 57, pp. 1511–1515, 2012.

[21] F. Sandblom and L. Svensson, "Moment estimation using a marginalized transform," *IEEE Transactions on Signal Processing*, vol. 60, pp. 6138–6150, 2012.

[22] A. O'Hagan, "Curve fitting and optimal design for prediction (with discussion)," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 40(1), pp. 1–42, 1978.

[23] ——, "Bayes-Hermite quadrature," *Journal of Statistical Planning and Inference*, vol. 29, pp. 245–260, 1991.

[24] T. P. Minka, "Deriving quadrature rules from Gaussian processes," Statistics Department, Carnegie Mellon University, Tech. Rep., 2000.

[25] M. A. Osborne, R. Garnett, S. J. Roberts, C. Hart, S. Aigrain, N. P. Gibson, and S. Aigrain, "Bayesian quadrature for ratios: Now with even more Bayesian quadrature," in *International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, 2012.

[26] M. Osborne, D. Duvenaud, R. Garnett, C. Rasmussen, S. Roberts, and Z. Ghahramani, "Active learning of model evidence using Bayesian quadrature," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 46–54.

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

[28] M. Deisenroth, R. Turner, M. Huber, U. Hanebeck, and C. Rasmussen, "Robust filtering and smoothing with Gaussian processes," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1865–1871, 2012.

[29] J. Hartikainen, "Sequential inference for latent temporal Gaussian process models," Doctoral dissertation, Aalto University, 2013.

[30] S. Särkkä, "Linear operators and stochastic partial differential equations in Gaussian process regression," in *Artificial Neural Networks and Machine Learning–ICANN 2011*. Springer, 2011, pp. 151–158.

[31] M. Alvarez and N. D. Lawrence, "Sparse convolved Gaussian processes for multi-output regression," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. The MIT Press, 2009, pp. 57–64.

[32] M. Alvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence, "Efficient multioutput Gaussian processes through variational inducing kernels," in *Proceedings of the 13th International Workshop on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterington, Eds., 2010, pp. 25–32.

[33] J. M. Hammersley, "Monte carlo methods for solving multivariable problems," *Annals of the New York Academy of Sciences*, vol. 86, no. 3, pp. 844–874, 1960.

[34] P. Malliavin, *Stochastic Analysis*, ser. Grundlehren der mathematischen Wissenschaften 313. Springer, 1997.