On the relation between Gaussian process quadratures and sigma-point methods

SIMO SÄRKKÄ
JOUNI HARTIKAINEN
LENNART SVENSSON
FREDRIK SANDBLOM

This article is concerned with Gaussian process quadratures, which are numerical integration methods based on Gaussian process regression methods, and sigma-point methods, which are used in advanced non-linear Kalman filtering and smoothing algorithms. We show that many sigma-point methods can be interpreted as Gaussian process quadrature based methods with suitably selected covariance functions. We show that this interpretation also extends to more general multivariate Gauss-Hermite integration methods and related spherical cubature rules. Additionally, we discuss different criteria for selecting the sigma-point locations: exactness of the integrals of multivariate polynomials up to a given order, minimum average error, and quasi-random point sets. The performance of the different methods is tested in numerical experiments.

Manuscript received November 10, 2015; released for publication November 26, 2015.

Refereeing of this contribution was handled by Ondrej Straka.

The first author is grateful to the Academy of Finland for financial support.

There are no conflict-of-interest or financial disclosure statements to be made at this time.

Authors' addresses: S. Särkkä, Aalto University, Rakentajanaukio 2 c, 02150 Espoo, Finland, (e-mail: simo.sarkka@aalto.fi). J. Hartikainen, Rocsole Ltd., 70150 Kuopio, Finland, (e-mail: jouni.hartikainen @rocsole.com). L. Svensson, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden, (e-mail: lennart.svensson@chalmers.se). F. Sandblom, Volvo Group Trucks Technology, SE-405 08 Gothenburg, Sweden, (e-mail: fredrik.sandblom@volvo.com).

1557-6418/16/$17.00

I. INTRODUCTION

Gaussian process quadratures [1]–[6] are methods to numerically compute integrals of the form

$$\mathcal{I}[\mathbf{g}] = \int \mathbf{g}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}, \tag{1}$$

where $\mathbf{g} : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a non-linear integrand function and $w(\mathbf{x})$ is a given, typically positive, weight function such that $\int w(\mathbf{x}) d\mathbf{x} < \infty$. In Gaussian process quadratures the function $\mathbf{g}(\mathbf{x})$ is approximated with a Gaussian process regressor [7] and the integral is approximated with that of the Gaussian process regressor.

Sigma-point methods [8]–[16] can be seen [17] as methods that approximate the above integrals via

$$\int \mathbf{g}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} \approx \sum_i W_i \mathbf{g}(\mathbf{x}_i), \tag{2}$$

where W_i are some predefined weights and $\mathbf{x}_i$ are the sigma-points. Typically the sigma-points and weights are selected such that when $\mathbf{g}$ is a multivariate polynomial up to a certain order, the approximation is exact.

A particularly useful class of methods is obtained when the weight function is selected to be a multivariate Gaussian density $w(\mathbf{x}) = \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})$. In the context of Gaussian process quadratures it then turns out that the integral of the Gaussian process regressor can be computed in closed form provided that the covariance function of the process is chosen to be a squared exponential [7], [18] (i.e., exponentiated quadratic). This kind of quadrature methods is also often referred to as Bayesian or Bayes-Hermite quadratures [2]. They are closely related to Gauss-Hermite 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 approximation instead [1]–[3]. Because Gaussian process regressors can be used to approximate a much larger class of functions than polynomial approximations [7], they can be expected to perform much better also in numerical integration.

The selection of a Gaussian weight function is also particularly useful in non-linear filtering and smoothing, because the equations of non-linear Gaussian (Kalman) filters and smoothers [17], [19]–[22] consist of Gaussian integrals of the above form and linear operations on vectors and matrices. The selection of different weights and sigma-points leads to different brands of approximate filters and smoothers [17]. For example, the Gauss-Hermite quadrature and cubature based filters and smoothers [21]–[25] are based on explicit numerical integration of the Gaussian integrals. The unscented transform based methods as well as other sigma-point methods [8]–[16] can also be retrospectively interpreted to belong to the class of Gaussian numerical integration based methods [23]. 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 [26] and Stirling's interpolation based methods [16], [27] can be seen as ways to approximate the integrand such that the Gaussian integral becomes tractable (cf. [17], [28]). The Fourier-Hermite series [29], Hermite polynomial [30] methods are also based on numerical approximation of the integrands. For more recent advances and applications of sigma-point methods in filtering and smoothing, see the articles [31]–[35] and the references therein.

The aim of this article is to provide a Gaussian process quadrature viewpoint to sigma-point methods and multivariate numerical integration methods for non-linear filtering and smoothing. The generalized viewpoint also leads to novel non-linear filtering and smoothing algorithms. We show that many sigma-point filtering and smoothing algorithms such as unscented Kalman filters and smoothers, cubature Kalman filters and smoothers, and Gauss-Hermite Kalman filters and smoothers can be seen as special cases of Gaussian process quadrature based methods with suitably chosen covariance functions. More generally, we show that many classical multivariate Gaussian quadrature methods, including Gauss-Hermite rules [36], and symmetric integration formulas [37] are special cases of the present methodology. We also discuss different criteria for selecting the sigma-point locations: exactness for multivariate polynomials up to a given order, minimum average error, and quasi-random point sets.

The combination of Gaussian process regressors with Bayesian filters has also been previously studied in [38] and [18]. In both of those works the idea is to use training data to form Gaussian process approximations to the dynamic and observation models. In [38], filtering in the resulting Gaussian process state-space model is done using approximate Bayesian filters such as unscented Kalman filters and particle filters, whereas in [18] the non-linear (Kalman) filtering and smoothing equations are computed via closed form formulas. The present approach and point of view is different, because we use Gaussian processes to approximate the integrals (quadratures) appearing in the filtering and smoothing equations. In practical point of view this roughly corresponds to locally retraining the Gaussian process regressor at every step using specifically designed training point locations.

The Gaussian process quadrature methodology used here can be seen to belong to a larger field of probabilistic numerics [39]–[41], where the underlying idea is to interpret numerical methods as instances of probabilistic inference. For example, numerical integration amounts to computing an estimate of the integral of a function given a finite number of function evaluations, whereas differential equation solvers estimate the ODE solution given a sequence of derivative evaluations, and optimization methods use local estimates of the target function to steer their iterations. Although in this article we only use probabilistic numerical integration, it is clear that probabilistic ODE solvers, optimization methods, and other probabilistic numerical methods would be useful in non-linear filtering and smoothing context as well.

This article is an extended version of the conference article [6], where we analyzed the use of Gaussian process quadratures in non-linear filtering and smoothing as well as their connection to the unscented transform and Gauss-Hermite quadratures. In this article, we deepen and sharpen the analysis of those connections and extend our analysis to a more general class of spherically symmetric integration rules. We also analyze different sigma-point selection schemes as well as provide more extensive set of numerical experiments.

II. BACKGROUND

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

Non-linear Gaussian (Kalman) filters and smoothers [17], [21]–[23] are methods that 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

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}, \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k, \end{aligned} \tag{3}$$

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 $\mathbf{q}_{k-1} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ 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 dynamics of the system and $\mathbf{h}(\cdot)$ models the mapping from the states to the measurements.

Non-linear Gaussian filters (see, e.g., [17], page 98) are general methods to produce Gaussian approximations to the filtering distributions:

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

Non-linear Gaussian smoothers (see, e.g., [17], page 154) are the corresponding methods to produce approximations to the smoothing distributions:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k^s, \mathbf{P}_k^s), \quad k = 1, 2, \ldots, T. \tag{5}$$

Both Gaussian filters and smoothers can be easily generalized to state-space models with non-additive noises (see [17]), but here we only consider the additive noise case.

B. Gaussian Integration and Sigma-Point Methods

Sigma-point filtering and smoothing methods can generally be described as methods that approximate the Gaussian integrals in the Gaussian filtering and smoothing equations (and in the Gaussian moment matching transform) as

$$\int \mathbf{g}(\mathbf{x}) \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) d\mathbf{x} \approx \sum_i W_i \mathbf{g}(\mathbf{x}_i), \tag{6}$$

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

$$\int \mathbf{g}(\mathbf{x})\mathrm{N}(\mathbf{x} \mid \mathbf{m},\mathbf{P})d\mathbf{x} = \int \underbrace{\mathbf{g}(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi})}_{\tilde{\mathbf{g}}(\boldsymbol{\xi})}\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}, \quad (7)$$

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

$$\int \tilde{\mathbf{g}}(\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi} \approx \sum_i W_i \tilde{\mathbf{g}}(\boldsymbol{\xi}_i), \quad (8)$$

thus leading to approximations of the form

$$\int \mathbf{g}(\mathbf{x})\mathrm{N}(\mathbf{x} \mid \mathbf{m},\mathbf{P})d\mathbf{x} \approx \sum_i W_i \mathbf{g}(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i). \quad (9)$$

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 [8] uses the following set of $2n+1$ weights (recall that n is the dimensionality of the state) and sigma-points:

$$W_0 = \frac{\kappa}{n+\kappa}, \quad W_i = \frac{1}{2(n+\kappa)}, \quad i = 1,\ldots,2n,$$
$$\boldsymbol{\xi}_i = \begin{cases} \mathbf{0}, & i = 0, \\ \sqrt{n+\kappa}\,\mathbf{e}_i, & i = 1,\ldots,n, \\ -\sqrt{n+\kappa}\,\mathbf{e}_{i-n}, & i = n+1,\ldots,2n. \end{cases} \quad (10)$$

where κ is a design parameter in the algorithm and $\mathbf{e}_i \in \mathbb{R}^n$ 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 that 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., [17]).

C. Gaussian Process Regression

Gaussian process quadrature [2], [3] is based on forming a Gaussian process (GP) regression [7] approximation to the integrand using pointwise evaluations and then integrating the approximation. In GP regression [7] the purpose is to predict the value of an unknown function

$$o = g(\mathbf{x}) \quad (11)$$

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 $g_\theta(\mathbf{x};\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^D$ 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:

$$g_K \sim \mathrm{GP}(0,K(\mathbf{x},\mathbf{x}')),$$
$$o_j = g_K(\mathbf{x}_j) + \epsilon_j, \quad \epsilon_j \sim \mathrm{N}(0,\sigma^2), \quad (12)$$

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

$$K(\mathbf{x},\mathbf{x}') = s^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x}-\mathbf{x}'\|^2\right), \quad (13)$$

where $s,\ell > 0$ are parameters of the covariance function (see [7]).

The GP regression equations can be derived as follows. Assume that we want to estimate the value of the noise-free 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

$$\begin{pmatrix} \mathbf{o} \\ g_K(\mathbf{x}) \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{K}+\sigma^2\mathbf{I} & \mathbf{k}(\mathbf{x}) \\ \mathbf{k}^{\mathrm{T}}(\mathbf{x}) & K(\mathbf{x},\mathbf{x}) \end{pmatrix}\right), \quad (14)$$

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, and $\mathbf{k}(\mathbf{x}) = [K(\mathbf{x},\mathbf{x}_i)]$ is the vector of cross covariances with the test point.

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

$$\mathrm{E}[g_K(\mathbf{x}) \mid \mathbf{o}] = \mathbf{k}^{\mathrm{T}}(\mathbf{x})(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{o}$$
$$\mathrm{Cov}[g_K(\mathbf{x}) \mid \mathbf{o}] = K(\mathbf{x},\mathbf{x}') - \mathbf{k}^{\mathrm{T}}(\mathbf{x})(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}'). \quad (15)$$

These are the Gaussian process regression equations in their typical form [7], in the special case where g is scalar. The extension to multiple output dimensions is conceptually straightforward (see, e.g., [7], [42]), but construction of the covariance functions as well as the practical computational methods tends to be complicated [43], [44]. 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 [2], [3] the basic idea is to approximate the integral of a given function g against a weight function $w(\mathbf{x})$, that is,

$$\mathcal{I}[g] = \int g(\mathbf{x})w(\mathbf{x})d\mathbf{x}, \quad (16)$$

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. Here we assume that g is scalar for simplicity as we can always take a vector function elementwise.

Gaussian process quadratures are related to a regression interpretation of classical Gaussian quadratures which means that we can interpret many of the classical methods as orthogonal polynomial approximations of the integrand evaluated at certain finite number of points [3]. The integral is then approximated by integrating the polynomial instead of the original function. However, the aim of Gaussian process quadrature is to get a good performance in average, whereas in classical polynomial quadratures the integration rule is designed to be exact for a limited class of (polynomial) functions. Still, these approaches are very much linked together [3].

Due to linearity of integration, the posterior mean of the integral of the Gaussian process regressor is given as

$$\mathrm{E}\left[\int g_K(\mathbf{x})w(\mathbf{x})d\mathbf{x} \mid \mathbf{o}\right] = \int \mathrm{E}[g_K(\mathbf{x}) \mid \mathbf{o}]w(\mathbf{x})d\mathbf{x}, \quad (17)$$

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 [2]–[5]. The posterior covariance of the approximation is

$$\begin{aligned}&\mathrm{Var}\left[\int g_K(\mathbf{x})w(\mathbf{x})d\mathbf{x} \mid \mathbf{o}\right] \\ &\quad = \iint \mathrm{Cov}[g_K(\mathbf{x}) \mid \mathbf{o}]w(\mathbf{x})d\mathbf{x}\,w(\mathbf{x}')d\mathbf{x}'. \quad (18)\end{aligned}$$

That is, when we approximate the integral (16) with the posterior mean we have

$$\int g(\mathbf{x})w(\mathbf{x})d\mathbf{x} \approx \left[\int \mathbf{k}^{\mathrm{T}}(\mathbf{x})w(\mathbf{x})d\mathbf{x}\right](\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{o}. \quad (19)$$

The posterior variance of the (scalar) integral is

$$\begin{aligned}&\mathrm{Var}\left[\int g_K(\mathbf{x})w(\mathbf{x})d\mathbf{x} \mid \mathbf{o}\right] \\ &\quad = \iint K(\mathbf{x},\mathbf{x}')w(\mathbf{x})d\mathbf{x}\,w(\mathbf{x}')d\mathbf{x}' \\ &\qquad - \left[\int \mathbf{k}^{\mathrm{T}}(\mathbf{x})w(\mathbf{x})d\mathbf{x}\right](\mathbf{K} + \sigma^2\mathbf{I})^{-1}\left[\int \mathbf{k}(\mathbf{x}')w(\mathbf{x}')d\mathbf{x}'\right]. \quad (20)\end{aligned}$$

In this article we are specifically interested in the case of Gaussian weight function, which then reduces the integral appearing in the above expressions (19) and (20) to

$$\left[\int \mathbf{k}^{\mathrm{T}}(\mathbf{x})w(\mathbf{x})d\mathbf{x}\right]_i = \int K(\mathbf{x},\mathbf{x}_i)\mathrm{N}(\mathbf{x} \mid \mathbf{m},\mathbf{P})d\mathbf{x}. \quad (21)$$

It is now easy to see that when the covariance function is a squared exponential $K(\mathbf{x},\mathbf{x}_i) = s^2\exp(-(2\ell^2)^{-1}\|\mathbf{x} - \mathbf{x}_i\|^2)$, this integral can be computed in closed form by using the computation rules for Gaussian distributions. Furthermore if the covariance function is a multivariate polynomial, then these integrals are given by the moments of the Gaussian distributions, which are also available in closed form.

III. GAUSSIAN PROCESS QUADRATURES FOR SIGMA-POINT FILTERING AND SMOOTHING

In this section we start by showing how Gaussian process quadratures (GPQ) can be seen as sigma-point methods and then introduce the Gaussian process transform (GPT). The Gaussian process transform then enables us to construct GPQ-based non-linear filters and smoothers analogously to [17].

A. GPQ as a sigma-point method

In this section the aim is to show how Gaussian process quadratures (GPQ) can be seen as sigma-point methods.

LEMMA III.1 *(GPQ as a sigma-point method). The Gaussian process quadrature (or Bayes-Hermite/Bayesian quadrature) can be seen as a sigma-point-type of integral approximation*

$$\int \mathbf{g}(\mathbf{x})\mathrm{N}(\mathbf{x} \mid \mathbf{m},\mathbf{P})d\mathbf{x} \approx \sum_{i=1}^{N} W_i\mathbf{g}(\mathbf{x}_i), \quad (22)$$

where $\mathbf{x}_i = \mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i$*, with the unit sigma-points* $\boldsymbol{\xi}_i$ *selected according to a predefined criterion, and the weights are determined by*

$$W_i = \left[\left(\int \mathbf{k}^{\mathrm{T}}(\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}\right)(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\right]_i, \quad (23)$$

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 of cross covariances. In principle, the choice of unit sigma-points above is completely free, but good choices of them are discussed in the following sections.*

PROOF Let us first use stochastic decoupling (7), which enables us to only consider unit-Gaussian integration formulas of the form (8). Because we can integrate vector functions element-by-element, without loss of generality we can assume that $g(\mathbf{x})$ is single-dimensional. Let us now model the function $\boldsymbol{\xi} \mapsto g(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi})$ as a Gaussian process g_K with a given covariance function $K(\boldsymbol{\xi},\boldsymbol{\xi}')$ and fix the training set for the GP regressor by selecting the points

$\boldsymbol{\xi}_i$, $i = 1,\ldots,N$, which also determines the corresponding points $\mathbf{x}_i = \mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i$ such that the training set is $\mathbf{o} = (g(\mathbf{x}_1),\ldots,g(\mathbf{x}_N))$. The GP approximation to the integral now follows from (19):

$$\begin{aligned}\int g(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid 0,\mathbf{I})d\boldsymbol{\xi} \\ \approx \left[\int \mathbf{k}^{\mathrm{T}}(\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid 0,\mathbf{I})d\boldsymbol{\xi}\right](\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{o}, \quad (24)\end{aligned}$$

which when simplified and applied to all the dimensions of $\mathbf{g}$ gives the result.

Note that above we actually assume that the stochastically-decoupled-function $\boldsymbol{\xi} \mapsto g(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi})$ instead of the original integrand $g(\mathbf{x})$ has the given covariance function. The reason for this modeling choice is that it enables us to decouple the mean and covariance from the integration formula and hence it is computationally beneficial. This also makes the result invariant to affine transformations of the state and it also has a property that the variability of the functions corresponds to the scale of the problem. However, on the other hand, one might argue that it is the function $g(\mathbf{x})$ which we should actually model and using the stochastically-decoupled-function is "wrong."

REMARK III.1 (Variance of GPQ). From Equation (20) we get that the component-wise variances of the Gaussian process quadrature approximation can be expressed as

$$\begin{aligned}V_j = &\iint K(\boldsymbol{\xi},\boldsymbol{\xi}')\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}\,\mathrm{N}(\boldsymbol{\xi}' \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}' \\ &- \int \mathbf{k}^{\mathrm{T}}(\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}(\mathbf{K} + \sigma^2\mathbf{I})^{-1} \\ &\times \int \mathbf{k}(\boldsymbol{\xi}')\mathrm{N}(\boldsymbol{\xi}' \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}'. \quad (25)\end{aligned}$$

Using the above integration approximations we can also define a general Gaussian process transform as follows. The reason for introducing the transform is that the corresponding approximate filters and smoothers can be readily constructed in terms of the transform (cf. [17]), which we will do in the next section.

ALGORITHM III.1 *(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*

$$\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix}\mathbf{m}\\ \boldsymbol{\mu}_{\mathrm{GP}}\end{pmatrix}, \begin{pmatrix}\mathbf{P} & \mathbf{C}_{\mathrm{GP}}\\ \mathbf{C}_{\mathrm{GP}}^{\mathrm{T}} & \mathbf{S}_{\mathrm{GP}}\end{pmatrix}\right), \quad (26)$$

where

$$\begin{aligned}\mathbf{x}_i &= \mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i,\\ \boldsymbol{\mu}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i\mathbf{g}(\mathbf{x}_i),\\ \mathbf{S}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i(\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}})(\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}})^{\mathrm{T}} + \mathbf{Q},\\ \mathbf{C}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i(\mathbf{x}_i - \mathbf{m})(\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}})^{\mathrm{T}}. \quad (27)\end{aligned}$$

Above, $\boldsymbol{\xi}_i$ *is some fixed set of sigma/training points and the weights are given by Equation (23) with some selected covariance function* $K(\boldsymbol{\xi},\boldsymbol{\xi}')$*.*

In this article, at least in the analytical results, we usually assume that the measurements are noise-free, that is, $\sigma^2 = 0$. This enables us to obtain analytically exact relationships with the classical quadrature methods. However, when using Gaussian process quadratures as numerical integration methods, it is often beneficial to have at least a small non-zero value for σ^2 in (23). This kind of "jitter" stabilizes numerics and can even be sometimes used to compensate for inaccuracies in modeling.

EXAMPLE III.1 (GPT with squared exponential kernel). Let us now consider $\xi \in \mathbb{R}$ and select the sigma-point locations to be the ones of unscented transform (10). With the squared exponential covariance function (13) and noise-free measurements ($\sigma^2 = 0$) we then get the weights:

$$W = \begin{pmatrix} \dfrac{\mathrm{e}^{-\frac{\kappa+1}{2(\ell^2+1)}}\left(\ell\mathrm{e}^{\frac{\kappa+1}{2(\ell^2+1)}} - 2\ell\mathrm{e}^{\frac{3(\kappa+1)}{2\ell^2}} + \ell\mathrm{e}^{\frac{\kappa+1}{2(\ell^2+1)}}\mathrm{e}^{\frac{2(\kappa+1)}{\ell^2}}\right)}{\sqrt{\ell^2+1}\left(\mathrm{e}^{\frac{\kappa+1}{\ell^2}} - 1\right)^2} \\ -\dfrac{\ell\mathrm{e}^{\frac{(2\ell^2+3)(\kappa+1)}{2\ell^2(\ell^2+1)}}\left(\mathrm{e}^{\frac{\kappa+1}{2(\ell^2+1)}} - \mathrm{e}^{\frac{\kappa+1}{2\ell^2}}\right)}{\sqrt{\ell^2+1}\left(\mathrm{e}^{\frac{\kappa+1}{\ell^2}} - 1\right)^2} \\ -\dfrac{\ell\mathrm{e}^{\frac{(2\ell^2+3)(\kappa+1)}{2\ell^2(\ell^2+1)}}\left(\mathrm{e}^{\frac{\kappa+1}{2(\ell^2+1)}} - \mathrm{e}^{\frac{\kappa+1}{2\ell^2}}\right)}{\sqrt{\ell^2+1}\left(\mathrm{e}^{\frac{\kappa+1}{\ell^2}} - 1\right)^2} \end{pmatrix}. \quad (28)$$

An interesting property is that in the limit $\ell \to \infty$ we get

$$\lim_{\ell\to\infty} W = \begin{pmatrix} \dfrac{\kappa}{\kappa+1} \\ \dfrac{1}{2(\kappa+1)} \\ \dfrac{1}{2(\kappa+1)} \end{pmatrix} \quad (29)$$

which are the unscented transform weights. We return to this relationship in Section IV-D.

B. GPQs in filtering and smoothing

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 [17], we can now formulate the following sigma-point filter for the model (3), which uses the unit sigma-points $\boldsymbol{\xi}_i$ and weights W_i defined by Algorithm III.1.

ALGORITHM III.2 *(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* $\mathbf{P}_k^-$*:*

$$\mathbf{m}_k^- = \sum_{i=1}^{N} W_i \hat{\mathcal{X}}_k^{(i)},$$

$$\mathbf{P}_k^- = \sum_{i=1}^{N} W_i (\hat{\mathcal{X}}_k^{(i)} - \mathbf{m}_k^-)(\hat{\mathcal{X}}_k^{(i)} - \mathbf{m}_k^-)^{\mathrm{T}} + \mathbf{Q}_{k-1}.$$

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 the 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* $\mathbf{C}_k$*:*

$$\boldsymbol{\mu}_k = \sum_{i=1}^{N} W_i \hat{\mathcal{Y}}_k^{(i)},$$

$$\mathbf{S}_k = \sum_{i=1}^{N} W_i (\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k)(\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k)^{\mathrm{T}} + \mathbf{R}_k,$$

$$\mathbf{C}_k = \sum_{i=1}^{N} W_i (\mathcal{X}_k^{-(i)} - \mathbf{m}_k^-)(\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k)^{\mathrm{T}}.$$

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* $\mathbf{y}_k$*:*

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

The result of the filter is a sequence of approximations

$$p(\mathbf{x}_k \mid \mathbf{y}_1,\ldots,\mathbf{y}_k) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k), \quad k = 1,2,\ldots,T. \tag{30}$$

Further following the line of thought in [17] we can formulate a sigma-point smoother using the unit sigma-points and weights from Algorithm III.1.

ALGORITHM III.3 *(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* $\mathbf{D}_{k+1}$*:*

$$\mathbf{m}_{k+1}^- = \sum_{i=1}^{N} W_i \hat{\mathcal{X}}_{k+1}^{(i)},$$

$$\mathbf{P}_{k+1}^- = \sum_{i=1}^{N} W_i (\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-)(\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-)^{\mathrm{T}} + \mathbf{Q}_k,$$

$$\mathbf{D}_{k+1} = \sum_{i=1}^{N} W_i (\mathcal{X}_k^{(i)} - \mathbf{m}_k)(\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-)^{\mathrm{T}}.$$

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

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

The approximations produced by the smoother are

$$p(\mathbf{x}_k \mid \mathbf{y}_1,\ldots,\mathbf{y}_T) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k^s, \mathbf{P}_k^s), \quad k = 1,2,\ldots,T. \tag{31}$$

Note that we could cope with non-additive noises in the model by using augmented forms of the above filters and smoothers as in [17]. The fixed-point and fixed-lag smoothers can also be derived analogously as was done in the same reference.

IV. SELECTION OF COVARIANCE FUNCTIONS AND SIGMA-POINT LOCATIONS

The accuracy of the Gaussian process quadrature method and hence the accuracy of the filtering and smoothing methods using it is affected by

1) the covariance function $K(\boldsymbol{\xi}, \boldsymbol{\xi}')$ used and
2) the sigma-point locations $\boldsymbol{\xi}_i$.

Once both of the above are fixed, the weights are determined by Equation (23). In this section we discuss certain useful choices of covariance functions as well as "optimal" choices of sigma-point locations for them. We also discuss the connection of the resulting methods with sigma-point methods such as unscented transforms and Gauss-Hermite quadratures.

A. Squared exponential and minimum variance point sets

In a machine learning context [7] the default choice for a covariance function of a Gaussian process is the squared exponential covariance function in Equation (13). What makes it convenient in Gaussian process quadratures is that the integral required for computing the weights in Equation (23) can be evaluated in closed form (cf. [3], [18]). It turns out that the posterior variance can be computed in closed form as well which is useful because for a given set of sigma-points we can immediately compute the expected error in the integral approximation (assuming that the integrand is indeed a GP)—this is possible because the variance does not depend on the observations at all.

One way to determine the sigma-point locations is to select them to minimize the posterior variance of the integral approximation [2], [3]. In our case this corresponds to minization of the variance in Equation (25) with respect to the points $\boldsymbol{\xi}_{1:N}$. Although the minimization is not possible in closed form, with a moderate N this optimization can be done numerically. Unfortunately, this numerical optimization problem is quite hard, because the optimum is far from being unique due to numerous symmetries appearing in the problem. Figure 1 shows examples of minimum variance point sets optimized by using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [45].

The squared exponential covariance function is not the only possible choice for a covariance function. From the machine learning context we could, for example, choose a Matérn covariance function or some of the scale-mixture-based covariance functions [7]. In that case the weight integral (23) becomes less trivial, but at least we always have a chance to precompute the weights using some (other) multivariate quadrature method. The sigma-point optimization could also be done similarly as for the squared exponential covariance function.

One potential disadvantage of using off-the-shelf covariance functions from machine learning is that they usually do not lead to filtering and smoothing methods which would give the exact result for linear state-space models. Recall that unscented Kalman filters and smoothers as well as Gauss-Hermite filters and smoothers do give the linear Kalman filter result when applied to a linear model. One way to diminish this issue is to use a covariance function which is formed as a sum of, for example, squared exponential covariance function and a suitable polynomial covariance function (see next section). Other ways include an explicit inclusion of the linear part into the regression model.

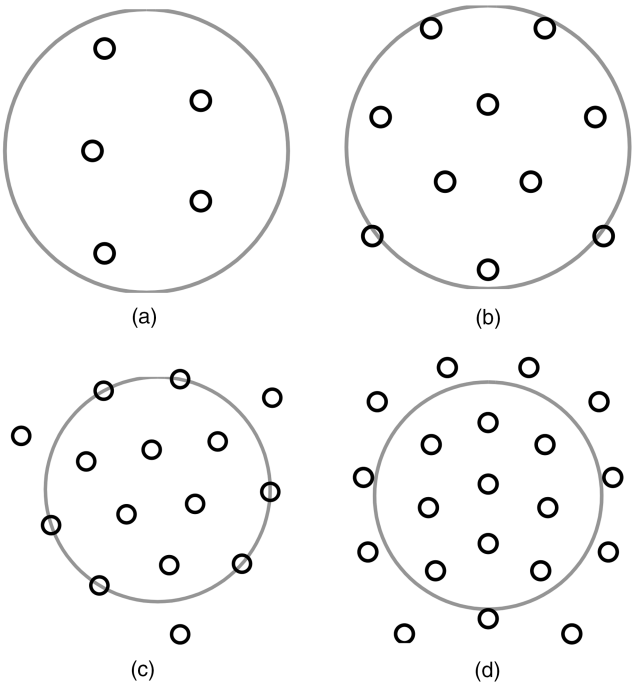


Fig. 1. Minimum variance (2d) point sets for the squared exponential covariance function. The gray circle is the unit circle depicted for visualization purposes. (a) 5 points. (b) 10 points. (c) 15 points. (d) 20 points.

B. UT and spherical cubature rules

In addition to the squared exponential covariance function, another useful class of covariance functions are polynomial covariance functions. They correspond to linear-in-parameters regression using polynomials as the regressor functions. It turns out that also for polynomial covariance functions we can compute the weights (23) in closed form. What is even more interesting is that the Gaussian process quadratures reduce to classical numerical integration methods. In this section we show that with certain selections of symmetric evaluation points we get a classical family of spherically symmetric integration methods of McNamee and Stenger [37] of which the unscented transform [8], [9] can be (retrospectively) seen as a special case [12]. More detailed information on the multivariate Hermite polynomials used below can be found in Appendix A.

THEOREM IV.1 *(UT covariance function). Assume that*

$$K(\boldsymbol{\xi},\boldsymbol{\xi}') = \sum_{q=0}^{3} \sum_{|J|=q} \sum_{p=0}^{3} \sum_{|I|=p} \frac{1}{\mathcal{I}!\mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} H_{\mathcal{I}}(\boldsymbol{\xi}) H_{\mathcal{J}}(\boldsymbol{\xi}'), \quad (32)$$

where $\lambda_{\mathcal{I},\mathcal{J}}$s form a positive definite covariance matrix and $H_{\mathcal{I}}(\boldsymbol{\xi})$ are multivariate Hermite polynomials (see Appendix A). If we now select the evaluation points as in UT (10), then the GPQ weights W_i become the UT weights. Furthermore, the posterior variance of the integral approximation is exactly zero.

PROOF The prior $g_K \sim \mathrm{GP}(0, K(\boldsymbol{\xi}, \boldsymbol{\xi}'))$ with the above covariance is equivalent to a parametric model of the form

$$g_K(\boldsymbol{\xi}) = \sum_{p=0}^{3} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} c_{\mathcal{I}} H_{\mathcal{I}}(\boldsymbol{\xi}), \tag{33}$$

where $c_{\mathcal{I}}$ are zero mean Gaussian random variables with the covariances $\lambda_{\mathcal{I},\mathcal{J}} = \mathrm{E}[c_{\mathcal{I}} c_{\mathcal{J}}]$. When the joint covariance matrix $\Lambda = [\lambda_{\mathcal{I},\mathcal{J}}]$ is non-singular, the posterior covariance of the integral being zero is equivalent to that the integral rule is exact for all functions of the form (33) with arbitrary coefficients. Note that we treat Λ as a covariance matrix despite formally being indexed by multi-indices. Clearly with the UT evaluations points, the UT weights are the unique ones that have this property (see, e.g., [17]) and hence the result follows.

Note that the above result also covers the cubature transform (CT), that is, the moment matching rule used in the cubature Kalman filter (CKF) and the smoother, because the transform is a special case of UT [17].

THEOREM IV.2 *(Higher order UT covariance function). Assume that*

$$K(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sum_{q=0}^{P} \sum_{|J|=q} \sum_{p=0}^{P} \sum_{|I|=p} \frac{1}{\mathcal{I}!\mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} H_{\mathcal{I}}(\boldsymbol{\xi}) H_{\mathcal{J}}(\boldsymbol{\xi}'). \tag{34}$$

If we select the evaluation points according to order $P = 5, 7, 9, \ldots$ rules in [37], we obtain the higher order integration formulas in [37], which are often referred to as the fifth order, seventh order, ninth order and higher order UTs.

PROOF The result follows analogously to the 3rd order case above.

Figure 2 shows two examples of unscented transform point sets, the 3rd and 5th order point sets (for 2 dimensions).

EXAMPLE IV.1 (Derivation of UT weights from GPQ). Let $\boldsymbol{\xi} \in \mathbb{R}^2$ and consider the GPQ with UT (10) sigma-points and the covariance function (32). With $\sigma = 0$ and $\lambda_{\mathcal{I},\mathcal{J}} = \delta_{\mathcal{I},\mathcal{J}}$ we then obtain the covariance matrix in (35).

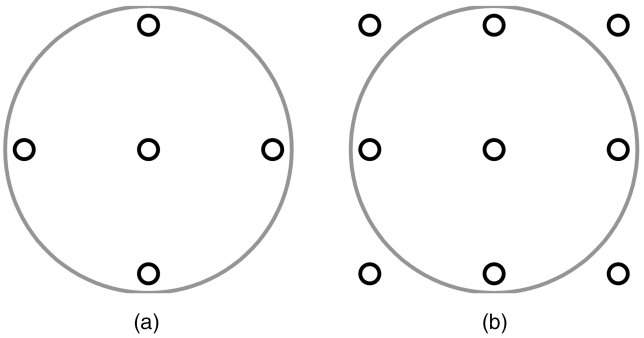


Fig. 2. Unscented transform point sets. (a) UT-3. (b) UT-5.

It also turns out that

$$\int \mathbf{k}^{\mathrm{T}}(\boldsymbol{\xi}) \mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0}, \mathbf{I}) d\boldsymbol{\xi} = (1 \cdots 1) \tag{36}$$

and finally

$$W_{0:4} = \left(\frac{\kappa}{\kappa+2} \quad \frac{1}{2(\kappa+2)} \quad \frac{1}{2(\kappa+2)} \quad \frac{1}{2(\kappa+2)} \quad \frac{1}{2(\kappa+2)} \right)^{\mathrm{T}}, \tag{37}$$

which are indeed the UT weights.

C. Multivariate Gauss-Hermite point sets

The multivariate Gauss-Hermite point sets (see, e.g., [17], [21]) of order P are exact for monomials of the form $x_1^{p_1} \times \cdots \times x_n^{p_n}$, where $p_i \leq 2P-1$ for $i = 1, \ldots, n$. This implies the following covariance function class.

THEOREM IV.3 *(Gauss-Hermite covariance function). Assume that*

$$K(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sum_{\max \mathcal{J} \leq 2P-1} \sum_{\max \mathcal{I} \leq 2P-1} \frac{1}{\mathcal{I}!\mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} H_{\mathcal{I}}(\boldsymbol{\xi}) H_{\mathcal{J}}(\boldsymbol{\xi}'), \tag{38}$$

where $\lambda_{\mathcal{I},\mathcal{J}}$s form a positive definite covariance matrix and $H_{\mathcal{I}}(\boldsymbol{\xi})$ are multivariate Hermite polynomials. If we now select the evaluation points to form a cartesian product of the roots of the Hermite polynomials of order P, then the GPQ weights W_i become the multivariate Gauss-Hermite quadrature weights. The posterior variance of the integral approximation is again exactly zero.

$$\mathbf{K} = \begin{pmatrix}
\frac{3}{2} & 1-\frac{\kappa}{4} & 1-\frac{\kappa}{4} & 1-\frac{\kappa}{4} & 1-\frac{\kappa}{4} \\
1-\frac{\kappa}{4} & \frac{\kappa^3}{36}+\frac{\kappa^2}{4}+\frac{13\kappa}{6}+\frac{91}{18} & \frac{1}{2}-\frac{\kappa}{2} & -\frac{\kappa^3}{36}+\frac{\kappa^2}{4}-\frac{7\kappa}{6}-\frac{37}{18} & \frac{1}{2}-\frac{\kappa}{2} \\
1-\frac{\kappa}{4} & \frac{1}{2}-\frac{\kappa}{2} & \frac{\kappa^3}{36}+\frac{\kappa^2}{4}+\frac{13\kappa}{6}+\frac{91}{18} & \frac{1}{2}-\frac{\kappa}{2} & -\frac{\kappa^3}{36}+\frac{\kappa^2}{4}-\frac{7\kappa}{6}-\frac{37}{18} \\
1-\frac{\kappa}{4} & -\frac{\kappa^3}{36}+\frac{\kappa^2}{4}-\frac{7\kappa}{6}-\frac{37}{18} & \frac{1}{2}-\frac{\kappa}{2} & \frac{\kappa^3}{36}+\frac{\kappa^2}{4}+\frac{13\kappa}{6}+\frac{91}{18} & \frac{1}{2}-\frac{\kappa}{2} \\
1-\frac{\kappa}{4} & \frac{1}{2}-\frac{\kappa}{2} & -\frac{\kappa^3}{36}+\frac{\kappa^2}{4}-\frac{7\kappa}{6}-\frac{37}{18} & \frac{1}{2}-\frac{\kappa}{2} & \frac{\kappa^3}{36}+\frac{\kappa^2}{4}+\frac{13\kappa}{6}+\frac{91}{18}
\end{pmatrix} \tag{35}$$

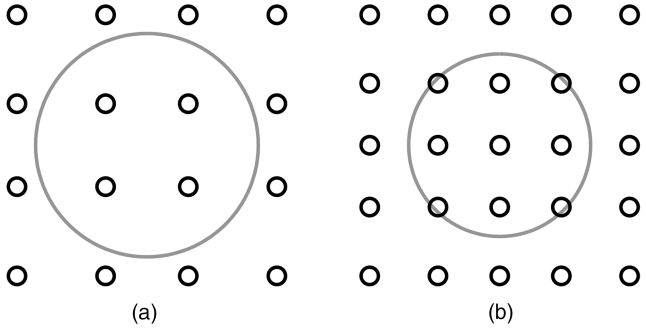


Fig. 3. Gauss-Hermite point sets. (a) GH-4. (b) GH-5.

PROOF Again the result follows from the equivalence of the polynomial approximations and polynomial covariance functions together with the uniqueness of the Gauss-Hermite rule for exact integration of this same function class.

Figure 3 shows 2d sigma-point sets formed as Cartesian products of two 4- and 5-point 1d Gauss-Hermite rules, respectively.

Even when we are using polynomial covariance functions, we are by no means restricted to using the specific points sets corresponding to the classical integration rules. However, given the order of the polynomial kernel and number of sigma-points they are also minimum variance points sets and hence good choices also in average—provided that the integrand is indeed a polynomial. In any case, for an arbitrary set of sigma-points we can use Equation (23) to give the corresponding minimum variance weights.

D. Connection between squared exponential and polynomial Gaussian process quadratures

As discussed in [3], the Gaussian process quadrature with squared exponential covariance function also has a strong connection with classical quadrature methods. This is because we can consider a set of damped polynomial basis functions of the form $\phi_i(\xi) = x^i \exp(-x^2/(2\ell^2))$, which at least informally speaking can be seen to converge to a polynomial basis when $\ell \to \infty$. We can now construct a family of random functions (Gaussian processes) of the form

$$g_\ell(x) = \sum_j c_j \phi_k(x) = \sum_j c_j x^j \exp\left(-\frac{x^2}{2\ell^2}\right), \qquad (39)$$

where $c_j \sim \mathrm{N}(0, (j! l^{2j})^{-1})$. The covariance function of this class is

$$\begin{aligned} K(x,y) &= \sum_i \frac{1}{i!\ell^{2i}} x^i \exp\left(-\frac{x^2}{2\ell^2}\right) y^i \exp\left(-\frac{y^2}{2\ell^2}\right) \\ &= \exp\left(\frac{xy}{\ell^2}\right) \exp\left(-\frac{x^2}{2\ell^2}\right) \exp\left(-\frac{y^2}{2\ell^2}\right) \\ &= \exp\left(-\frac{(x-y)^2}{2\ell^2}\right), \end{aligned} \qquad (40)$$

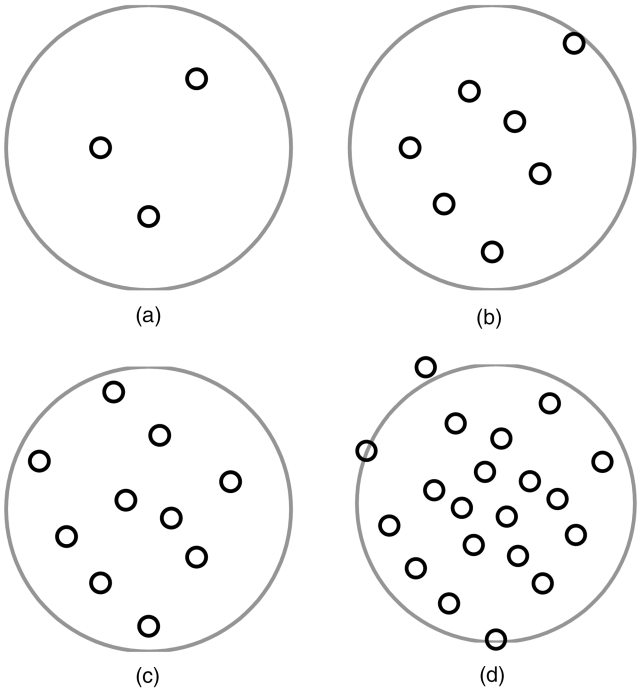


Fig. 4. Hammersley point sets. (a) 3 points. (b) 7 points. (c) 10 points. (d) 20 points.

which is the squared exponential covariance function.

Based on the above, Minka [3] argued (although did not formally prove) that GPQs with squared exponential covariance functions should converge to the classical quadratures. This argument is indeed backed up by our analytical example in Example III.1 where this covergence indeed happens.

E. Random and quasi-random point sets

Recall that one way to approximate the expectation of $\mathbf{g}(\boldsymbol{\xi})$ over a Gaussian distribution $\mathrm{N}(\mathbf{0}, \mathbf{I})$ is to use Monte Carlo integration. In that method we simply draw N samples from the Gaussian distribution $\xi_i \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$ and use them as sigma-points. The classical Monte Carlo approximation to the integral would now correspond to setting $W_i = 1/N$. Alternatively, we could use these random points as sigma-points and evaluate their weights by Equation (23). This leads to an approximation, which is sometimes called the Bayesian Monte Carlo approximation [46], [47].

Instead of sampling from the normal distribution, we can also use quasi-random points sets such as the Hammersley point sets [48], [49]. These are points sets which are designed to give a smaller error in average than random points. The classical method would correspond to setting all weights to $W_i = 1/N$, but again, we can also use Equation (23) to evaluate the weights for the GP quadrature. This corresponds to a "Bayesian quasi Monte Carlo" approximation to the integral. Some examples of Hammersley point sets are shown in Figure 4.

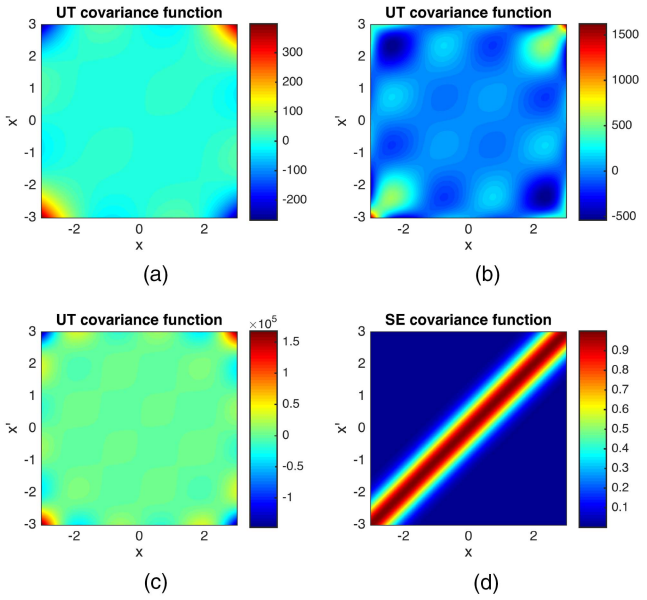


Fig. 5. Covariance functions corresponding to different orders of unscented transforms (UT) and the squared exponential (SE) covariance function ($s = 1$, $\ell = 1/2$) for a single-input scalar-valued Gaussian process. (a) UT-3. (b) UT-5. (c) UT-7. (d) SE.

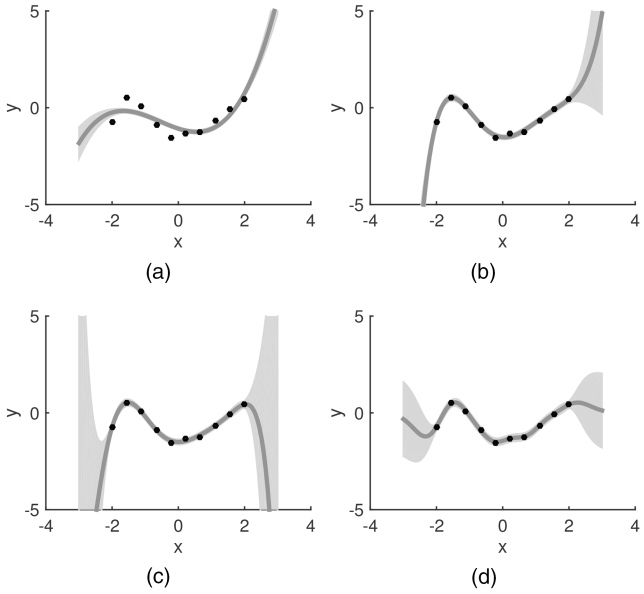


Fig. 6. Regression with covariance functions for UT and SE. (a) UT-3. (b) UT-5. (c) UT-7. (d) SE.

V. NUMERICAL RESULTS

A. Covariance functions and regression implied by unscented transform

The unscented transform covariance functions of orders 3–7 (see Theorems IV.1 and IV.2) and the exponentiated quadratic (i.e., the squared exponential, SE) covariance function (Eq. (13)) are illustrated in Fig. 5. The polynomial nature of the unscented transform (UT) covariance function can be clearly seen in the figures—the UT covariance function as such does not have such a simple local-correlation-interpretation as the SE covariance function has, because the UT covariance functions simply blow up polynomially when moving away from the diagonal.

The corresponding Gaussian process regression results on random data are illustrated in Fig. 6. The polynomial nature of the unscented transform can be clearly seen in the figures. The Gaussian process prediction with the unscented transform covariance function has a clear polynomial shape as expected. Clearly the polynomial fit has less flexibility to explain the data than the exponentiated quadratic fit although the flexibility certainly grows with the polynomial (and thus UT) order.

B. Illustrative high-dimensional example

We use the same test case as in Section VIII.A. of [24], that is, the computation of the first two moments of the function $y(\mathbf{x}) = (\sqrt{1 + \mathbf{x}^{\mathrm{T}}\mathbf{x}})^p$ for $p = 1, -2, -3, -5$. We thus aim to approximate the following integrals:

$$\mathrm{E}[y(\mathbf{x})] = \int \left(\sqrt{1 + \mathbf{x}^{\mathrm{T}}\mathbf{x}}\right)^p \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) d\mathbf{x}, \quad (41)$$

$$\mathrm{E}[y^2(\mathbf{x})] = \int (1 + \mathbf{x}^{\mathrm{T}}\mathbf{x})^p \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) d\mathbf{x}. \quad (42)$$

Figure 7 shows the result of using the following methods as function of the state-dimensionality:

- *Cubature:* The 3rd order spherical cubature sigma-points ($2n$ points) with the standard integration weights.
- *GPQ-Cubature:* The Gaussian process quadrature with SE covariance function and the 3rd order spherical cubature sigma-points above.
- *GPQ-Hammersley:* The Gaussian process quadrature with SE covariance and $2n$ Hammersley points.

The 3rd spherical cubature points refer to the integration rule proposed in [37], which was also used in the cubature Kalman filter (CKF) in [24]. In the rule, the sigma-points are placed to the intersections of coordinate axes with the origin-centered n-dimensional hypersphere of radius $\sqrt{n}$. Following [24] we measured the accuracy of the methods by computing accurate mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$ via Monte Carlo sampling and by comparing it to the approximate means $\mathbf{m}_1$ and covariances $\boldsymbol{\Sigma}_1$ using the following KL-divergence for two Gaussian distributions:

$$\mathrm{KL}[\mathrm{N}_0 \| \mathrm{N}_1] = \frac{1}{2}\left\{ \mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - n + \log\left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\right)\right\}. \quad (43)$$

The results in Figure 7 show that the GPQ quite consistently gives a bit lower KL-divergence and hence better result than the plain cubature when the cubature points are used. When Hammersley point sets are used, the results vary a bit more: with small state dimensions the results are slightly worse than with the cubature points. When $p \neq 1$, the Hammersley results are much better

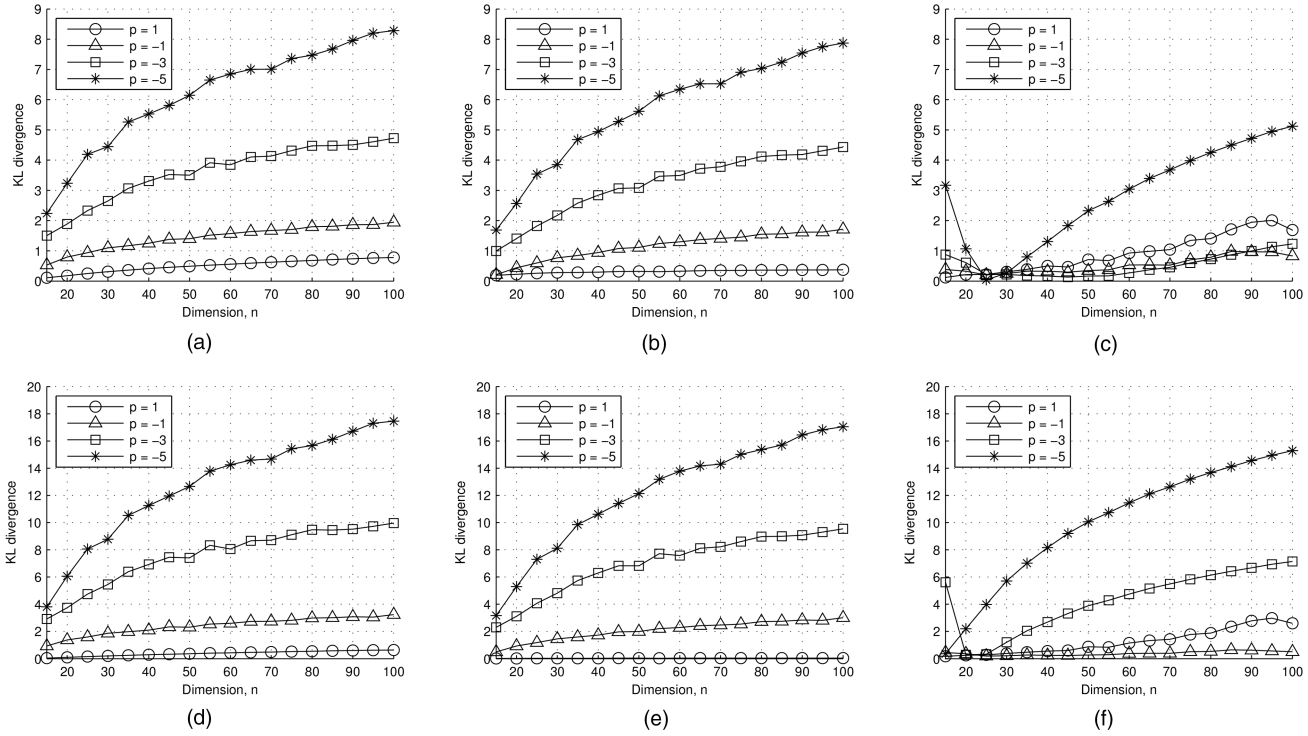


Fig. 7. Comparison of different methods in computing the moment integrals used in [24] for illustrating the performance of the cubature rule. It can be seen that the GPQ methods outperform the cubature rule in most of the cases. (a) Cubature for (41). (b) GPQ-Cubature for (41). (c) GPQ-Hammersley for (41). (d) Cubature for (42). (e) GPQ-Cubature for (42). (f) GPQ-Hammersley for (42).

in high dimensions whereas with $p = 1$ the results are worse than with the cubature point sets.

C. 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:

$$x_k = \frac{1}{2}x_{k-1} + 25\frac{x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + q_{k-1},$$
$$y_k = \frac{1}{20}x_k^2 + r_k, \tag{44}$$

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 (GPKFU/GPRTSU) and cubature points (GPKFC/GPRTSC); with Hammersley point sets of sizes 3, 7, and 10 (GPKFH/GPRTSH); and with minimum variance points sets of sizes 3, 7, and 10 (GPKFO/GPRTSO). 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 together with single standard derivation bars are shown in Figures 8 and 9. As can be seen in the figures, with 7 and 10 points the

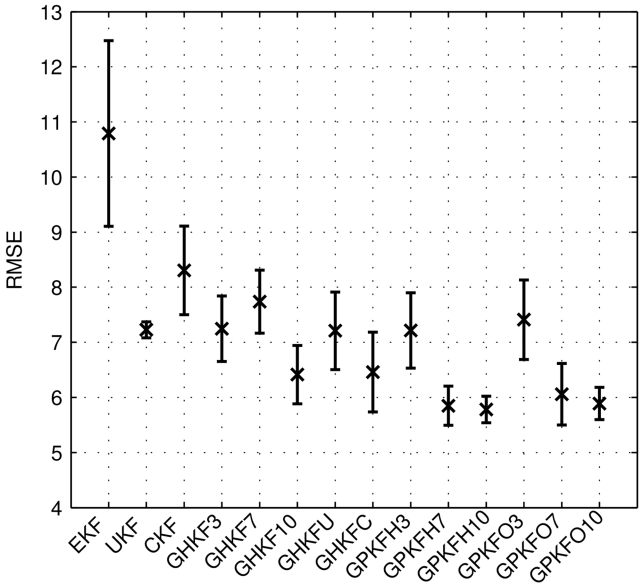


Fig. 8. RMSE results of filters in the UNGM experiment.

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 unscented RTS smoother come close.

D. Bearings only target tracking

In this section we evaluate the methods in the bearings only target tracking problem with a coordinated-turn dynamic model, which was also considered in Sec-

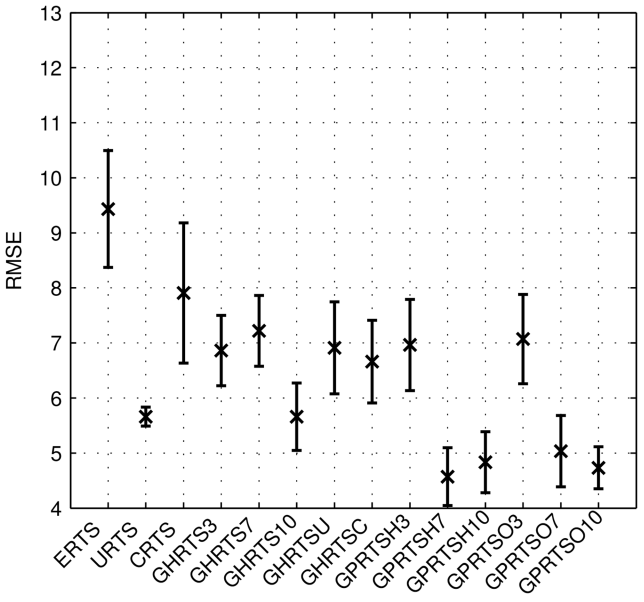


Fig. 9. RMSE results of smoothers in the UNGM experiment.

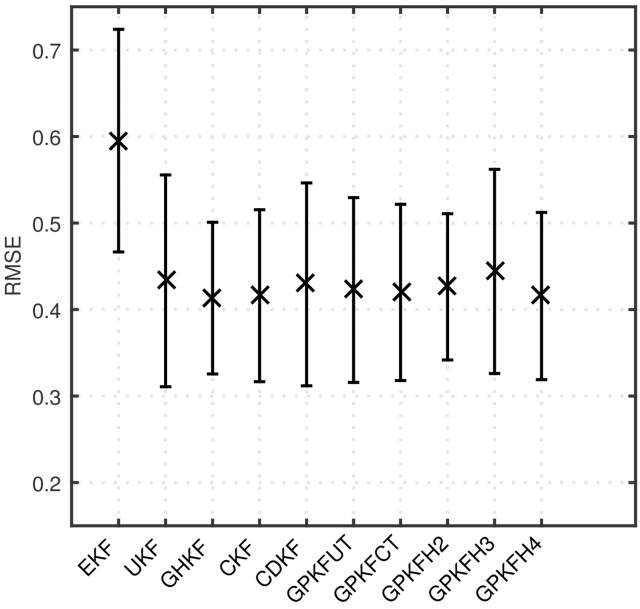


Fig. 10. Position RMSE results of filters in the bearings only experiment.

tion III.A of the article [22]. The non-linear dynamic model is

$$\mathbf{x}_k = \begin{pmatrix} 1 & \frac{\sin(\omega_k \Delta t)}{\omega} & 0 & -\left(\frac{1-\cos(\omega_k \Delta t)}{\omega}\right) & 0 \\ 0 & \cos(\omega_k \Delta t) & 0 & -\sin(\omega_k \Delta t) & 0 \\ 0 & \frac{1-\cos(\omega_k \Delta t)}{\omega_k} & 1 & \frac{\sin(\omega \Delta t)}{\omega} & 0 \\ 0 & \sin(\omega_k \Delta t) & 0 & \cos(\omega_k \Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \mathbf{x}_{k-1} + \mathbf{q}_{k-1}, \tag{45}$$

where the state of the target is $\mathbf{x} = (x_1, \dot{x}_1, x_2, \dot{x}_2, \omega)$, and x_1, x_2 are the coordinates and $\dot{x}_1, \dot{x}_2$ are the velocities in two dimensional space. The time step size is set to $\Delta t = 1$ s and the covariance of the process noise $q_k \sim N(0, Q)$ is

$$Q = \begin{pmatrix} q_1 \frac{\Delta t3}{3} & q_1 \frac{\Delta t2}{2} & 0 & 0 & 0 \\ q_1 \frac{\Delta t2}{2} & q_1 \Delta t & 0 & 0 & 0 \\ 0 & 0 & q_1 \frac{\Delta t3}{3} & q_1 \frac{\Delta t2}{2} & 0 \\ 0 & 0 & q_1 \frac{\Delta t2}{2} & q_1 \Delta t & 0 \\ 0 & 0 & 0 & 0 & q_2 \Delta t \end{pmatrix}, \tag{46}$$

where we used $q_1 = 0.1$ m^{2}s^{-3} and $q_2 = 1.75 \times 10^{-4}$ s^{-3}.

In the simulation setup we have four sensors measuring the angles θ between the target and the sensors. The non-linear measurement model for sensor i can be written as

$$\theta^i = \arctan\left(\frac{x_2 - s_2^i}{x_1 - s_1^i}\right) + r^i, \tag{47}$$

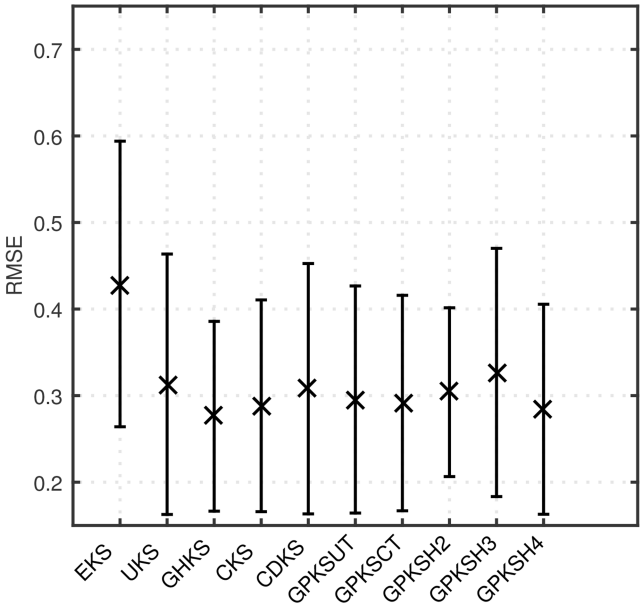


Fig. 11. Position RMSE results of smoothers in the bearings only experiment.

where (s_1^i, s_2^i) is the position of the sensor i in two dimensions, and $r^i \sim N(0, \sigma_\theta^2)$ is the measurement noise. The used parameters were the same as in the article [22].

The RMSE results for the position errors are shown in Figures 10 and 11. Clearly all of the sigma-point methods outperform the Taylor series based methods (EKF/EKS). However, the performances of all the sigma-point methods are very similar: also the Gaussian process quadrature methods give very similar results to the other sigma-point methods. There is a small dip in the errors at the Gauss-Hermite based methods as well as in the highest order Hammersley GPQ method, but

practically the performance of all the sigma-point methods is the same.

VI. CONCLUSION

In this article we have provided a Gaussian process quadrature viewpoint to sigma-point methods and multivariate numerical integration methods for non-linear filtering and smoothing. Using this viewpoint, we have also developed new Gaussian process quadrature based non-linear Kalman filtering and smoothing methods and analyzed their relationship with other sigma-point filters and smoothers. We have also discussed the selection of the evaluation points for the quadratures with respect to different criteria: exactness for multivariate polynomials up to a given order, minimum average error, and quasi-random point sets. We have shown that with suitable selections of (polynomial) covariance functions for the Gaussian processes the filters and smoothers reduce to unscented Kalman filters of different orders, as well as to Gauss-Hermite Kalman filters and smoothers. By numerical experiments we have also shown that the Gaussian process quadrature rules as well as the corresponding filters and smoothers often outperform previously proposed (polynomial) integration rules and sigma-point filters and smoothers.

At this point it is useful to reiterate where the relation of Gaussian process quadratures (GPQs) and sigma-point methods actually originates from. First of all, sigma-point filtering and smoothing methods can be seen as multivariate (classical) quadrature approximations to formal Gaussian (Kalman) filtering and smoothing equations. We also know that classical quadratures can be seen as methods that integrate a polynomial approximant of the function instead of the function itself. From probabilistic numerics we know that Gaussian process (i.e., Bayes-Hermite) quadrature corresponds to integrating a Gaussian process approximant of the function. Now the "kernel trick" tells us that polynomial interpolants can be converted into Gaussian process regressors by using a suitable polynomial covariance function. This implies that using a suitable polynomial covariance function in GPQ approximation of Gaussian (Kalman) filtering and smoothing equations will give us back the conventional sigma-point methods. The known limit results of GPQs converging to classical quadratures also directly translate to convergence of the GPQ based filters and smoothers to the conventional sigma-point methods.

APPENDIX A FOURIER-HERMITE SERIES

Fourier-Hermite series (see, e.g., [50]) are orthogonal polynomial series in a Hilbert space, where the inner product is defined via an expectation of the 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 [17], [29], [30].

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

$$\langle f, g \rangle = \int f(\mathbf{x}) g(\mathbf{x}) \mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I}) d\mathbf{x}. \tag{48}$$

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 [50].

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

$$H_{\mathcal{I}}(\mathbf{x}) = H_{i_1}(x_1) \times \cdots \times H_{i_n}(x_n), \tag{49}$$

which is a product of univariate Hermite polynomials

$$H_p(x) = (-1)^p \exp(x^2/2) \frac{d^p}{dx^p} \exp(-x^2/2). \tag{50}$$

The orthogonality property can now be expressed as

$$\langle H_{\mathcal{I}}, H_{\mathcal{J}} \rangle = \begin{cases} \mathcal{I}!, & \text{if } \mathcal{I} = \mathcal{J} \\ 0, & \text{otherwise,} \end{cases} \tag{51}$$

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 [50]

$$g(\mathbf{x}) = \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} c_{\mathcal{I}} H_{\mathcal{I}}(\mathbf{x}), \tag{52}$$

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

Consider a Gaussian process $g_G(\mathbf{x})$ that 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:

$$g_G(\mathbf{x}) = \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} \tilde{c}_{\mathcal{I}} H_{\mathcal{I}}(\mathbf{x}), \tag{53}$$

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

$$\begin{aligned} \mathrm{E}[\tilde{c}_{\mathcal{I}} \tilde{c}_{\mathcal{J}}] &= \mathrm{E}[\langle H_{\mathcal{I}}, g_G \rangle \langle H_{\mathcal{J}}, g_G \rangle] \\ &= \iint H_{\mathcal{I}}(\mathbf{x}) K(\mathbf{x}, \mathbf{x}') H_{\mathcal{J}}(\mathbf{x}') \\ &\quad \times \mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I}) \mathrm{N}(\mathbf{x}' \mid \mathbf{0}, \mathbf{I}) d\mathbf{x} d\mathbf{x}'. \end{aligned} \tag{54}$$

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

$$K(\mathbf{x},\mathbf{x}') = \sum_{q=0}^{\infty} \sum_{|\mathcal{J}|=q} \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!\mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} H_{\mathcal{I}}(\mathbf{x}) H_{\mathcal{J}}(\mathbf{x}'). \tag{55}$$

REFERENCES

[1] 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.

[2] ———
Bayes-Hermite quadrature,
Journal of Statistical Planning and Inference, vol. 29, pp. 245–260, 1991.

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

[4] 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.

[5] 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.

[6] S. Särkkä, J. Hartikainen, L. Svensson, and F. Sandblom
Gaussian process quadratures in nonlinear sigma-point filtering and smoothing,
in *Proceedings of FUSION 2014*, 2014.

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

[8] 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.

[9] 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, 2000.

[10] 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.

[11] 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, 2004.

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

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

[14] 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.

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

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

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

[18] 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.

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

[20] ———
Numerical approximations to optimal nonlinear filters,
in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovskiĭ, Eds. Oxford, 2011, ch. 28.

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

[22] 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, 2010.

[23] 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.

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

[25] ———
Cubature Kalman smoothers,
Automatica, vol. 47, no. 10, pp. 2245–2250, 2011.

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

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

[28] F. Gustafsson and G. Hendeby
Some relations between extended and unscented Kalman filters,
IEEE Transactions on Signal Processing, vol. 60, no. 2, pp. 545–555, 2012.

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

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

[31] J. Duník, O. Straka, and M. Šimandl
Stochastic integration filter,
IEEE Transactions on Automatic Control, vol. 58, no. 6, pp. 1561–1566, 2013.

[32] O. Straka, J. Duník, and M. Šimandl
Unscented Kalman filter with advanced adaptation of scaling parameter,
Automatica, vol. 50, no. 10, pp. 2657–2664, 2014.

[33] J. Kokkala, A. Solin, and S. Särkkä
Expectation maximization based parameter estimation by sigma-point and particle smoothing,
in *Proceedings of FUSION 2014*, 2014.

[34] J. Duník, O. Straka, M. Šimandl, and E. Blasch
Random-point-based filters: analysis and comparison in target tracking,
IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 2, pp. 1403–1421, 2015.

[35] A. Garcia-Fernandez, L. Svensson, M. Morelande, and S. Särkkä
Posterior linearization filter: Principles and implementation using sigma points,
IEEE Transactions on Signal Processing, vol. 63, no. 20, pp. 5561–5573, 2015.

[36] P. Davis and P. Rabinowitz
Methods of Numerical Integration,
2nd ed., ser. Computer science and applied mathematics. Academic Press, 1984.

[37] J. McNamee and F. Stenger
Construction of fully symmetric numerical integration formulas,
Numerische Mathematik, vol. 10, pp. 327–344, 1967.

[38] 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.

[39] P. Diaconis
Bayesian numerical analysis,
Statistical decision theory and related topics IV, vol. 1, pp. 163–175, 1988.

[40] A. O'Hagan
Some Bayesian numerical analysis,
Bayesian Statistics, vol. 4, pp. 345–363, 1992.

[41] P. Hennig, M. A. Osborne, and M. Girolami
Probabilistic numerics and uncertainty in computations,
Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 471, no. 2179, 2015.

[42] 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.

[43] 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.

[44] 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.

[45] R. Fletcher
Practical methods of optimization,
2nd ed. John Wiley & Sons, 1987.

[46] A. O'Hagan
Monte Carlo is fundamentally unsound,
Journal of the Royal Statistical Society. Series D (The Statistician), vol. 36, pp. 247–249, 1987.

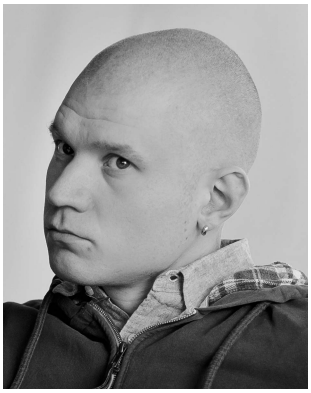
[47] Z. Ghahramani and C. E. Rasmussen
Bayesian Monte Carlo,
in *Advances in neural information processing systems*, 2002, pp. 489–496.

[48] 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.

[49] J. M. Hammersley and D. C. Handscomb
Monte Carlo Methods.
Fletcher & Son, 1964.

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

Simo Särkkä received his M.Sc. degree in engineering physics and mathematics, and D.Sc. degree in electrical and communications engineering from Helsinki University of Technology, Espoo, Finland, in 2000 and 2006, respectively. Currently he is an Associate Professor and Academy Research Fellow with Aalto University, and Technical Advisor and Director of IndoorAtlas Ltd. His research interests are in multi-sensor data processing systems with applications in location sensing, health technology, machine learning, inverse problems, and brain imaging.



Jouni Hartikainen received his M.Sc. degree in computer science and engineering, and D.Sc. (Ph.D.) degree in computational engineering from Aalto University, Espoo, Finland, in 2008 and 2013, respectively. Currently, he is a Senior Technology Manager with Rocsole Ltd. and his research interests are in Bayesian methods for dynamic inverse problems in industrial processes.

Lennart Svensson was born in Älvängen, Sweden in 1976. He received the M.S. degree in electrical engineering in 1999 and the Ph.D. degree in 2004, both from Chalmers University of Technology, Gothenburg, Sweden. He is currently Associate Professor at the Signal Processing group, again at Chalmers University of Technology. His main research interests include machine learning and Bayesian inference in general, and nonlinear filtering and tracking in particular.

Fredrik Sandblom was born in Mölndal, Sweden in 1979. He received the M.Sc. and Ph.D. degrees from Chalmers University of Technology in Gothenburg, Sweden, in 2004 and 2011 respectively. Since 2005 he has been with the Volvo group, working with research and development of active safety systems. His research interests concern system design, object tracking and sensor data fusion; particularly methods for estimating statistical moments and their application to recursive filtering.