Improved calibration of numerical integration error in sigma-point filters

Jakub Prüher, Toni Karvonen, Chris J. Oates, Ondřej Straka, and Simo Särkkä

This article has been accepted for publication as a technical note in *IEEE Transactions on Automatic Control*. When citing this article, use the following reference:

> J. Prüher, T. Karvonen, C. J. Oates, O. Straka, and S. Särkkä, "Improved calibration of numerical integration error in sigma-point filters," *IEEE Transactions on Automatic Control*, to appear, 2021.

Improved Calibration of Numerical Integration Error in Sigma-Point Filters

Jakub Prüher, Toni Karvonen, Chris J. Oates, Ondřej Straka and Simo Särkkä

***Abstract*—The sigma-point filters, such as the UKF, are popular alternatives to the ubiquitous EKF. The classical quadrature rules used in the sigma-point filters are motivated via polynomial approximation of the integrand, however in the applied context these assumptions cannot always be justified. As a result, quadrature error can introduce bias into estimated moments, for which there is no compensatory mechanism in the classical sigma-point filters. This can lead in turn to estimates and predictions that are poorly calibrated. In this article, we investigate the Bayes–Sard quadrature method in the context of sigma-point filters, which enables uncertainty due to quadrature error to be formalised within a probabilistic model. Our first contribution is to derive the well-known classical quadratures as special cases of the Bayes–Sard quadrature method. Based on this, a general-purpose moment transform is developed and utilised in the design of novel sigma-point filter, which explicitly accounts for the additional uncertainty due to quadrature error.**

***Index Terms*—Kalman filters, Bayesian quadrature, quantification of uncertainty, sigma-points, Gaussian processes.**

I. Introduction

THIS article is concerned with quantification of uncertainty associated with sigma-point approximations, which are widely employed in nonlinear filtering algorithms, such as the unscented Kalman filter (UKF). The goal of filtering algorithms is to estimate the state of a dynamical stochastic system based on all measurements obtained until the present. The applications of filters are manifold, ranging from global positioning [1], object tracking [2], [3], simultaneous localization and mapping [4] to weather forecasting [5] and finance [6].

Instead of keeping track of the whole state posterior, the sigma-point filters only work with mean and covariance of the state and the measurement. For nonlinear systems and/or measurements, the moments are defined by intractable integrals that have to be approximated using numerical quadratures, also known as the sigma-point rules (which is where the filters get their name). The classical quadrature rules, such as the Gauss–Hermite rule, are designed with the assumption that the nonlinear integrand is well-approximated with a polynomial of a given maximal degree. Since these assumptions are almost never met in practice, there will always be a quadrature error involved. Standard sigma-point filters do not attempt to compensate for this source of error, and in practice this can lead to estimates and predictions that are biased and overconfident [7], [8].

The *Bayesian quadrature* (BQ) has in recent years received much attention in the probabilistic numerics community [9]–[12]. The BQ approach posits that the integrand can be modelled by a stochastic process defined on the domain of integration. This model is subsequently refined by conditioning on point-wise evaluations of the integrand which induces a posterior distribution over the value of the integral. The posterior mean of this distribution is point estimate of the value of the integral while the posterior variance quantifies the integration error.

Applications of the BQ in nonlinear filtering have appeared previously in [13], [14] with encouraging results. These BQ-based filters do not generally coincide with any classical sigma-point filter, such as the UKF or Gauss–Hermite Kalman filter (GHKF), and tend to be rather sensitive to specification of the stochastic process model for the integrand. It has been shown that classical sigma-point rules can be cast as *degenerate* BQ rules [9], [13]. This is to say that the variance associated to the integral vanishes, being thus of no use in modelling integration error.

In this article we utilise the recently proposed Bayes–Sard quadrature [15] for the design of novel sigma-point filters, which can be viewed as probabilistic versions of the well-known sigma-point filters. Namely, under certain conditions, the Bayes–Sard quadrature allows us to recover the classical sigma-point rules and at the same time endow the sigma-point rule with non-degenerate probabilistic output. We thus obtain versions of standard sigma-point filters that are, to some extent, capable of accounting for numerical integration error in filtering by inflating the error covariance. In some cases, such covariance inflation is known to improve stability of nonlinear Kalman filters; see for instance [16, Remark 1], [17, Section 3.3], and [18, Section V.C].

The rest of the article is structured as follows. In Section II, we formally outline the nonlinear filtering problem and the nature of sigma-point approximations. Section III identifies the moment transformation problem as the central issue in sigma-point filtering and describes the structure of sigma-point moment transforms. The Bayes–Sard quadrature is formalised in Section IV, which is later used in Section V to design the Bayes–Sard quadrature moment transform. Section VII concludes the article.

II. Sigma-Point Filtering

This section is devoted to the sigma-point filters, which are a subset of nonlinear filtering algorithms characterised by their

J. Prüher and O. Straka are with Department of Cybernetics, University of West Bohemia, Czech Republic.

T. Karvonen is with Department of Electrical Engineering and Automation, Aalto University, Finland and the Alan Turing Institute, United Kingdom

S. Särkkä is with Department of Electrical Engineering and Automation, Aalto University, Finland.

C. J. Oates is with School of Mathematics, Statistics & Physics, Newcastle University and the Alan Turing Institute, United Kingdom.

reliance on a Gaussian approximation together with a numerical quadrature method. Let the stochastic dynamical system and the process by which its state is observed be described by the state-space model

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}, \tag{1}$$
$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k, \tag{2}$$

where the function $\mathbf{f} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ is the system dynamics, $\mathbf{h} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ is the measurement model, $\mathbf{x}_k \in \mathbb{R}^{d_x}$ is the latent state vector and $\mathbf{z}_k \in \mathbb{R}^{d_z}$ is the measurement vector. Both the process noise $\mathbf{q}_{k-1} \sim \mathrm{N}(\mathbf{0},\, \mathbf{Q})$ and the measurement noise $\mathbf{r}_k \sim \mathrm{N}(\mathbf{0},\, \mathbf{R})$ are zero-mean white Gaussian sequences, independent of each other and independent of the system initial condition $\mathbf{x}_0 \sim \mathrm{N}(\mathbf{m}_0^x,\, \mathbf{P}_0^x)$.

The Bayesian formulation of the filtering problem can be summarized by the following two general relations. The state posterior is $p(\,\mathbf{x}_k \,|\, \mathbf{z}_{1:k}\,) \propto p(\,\mathbf{z}_k \,|\, \mathbf{x}_k\,)p(\,\mathbf{x}_k \,|\, \mathbf{z}_{1:k-1}\,)$, where the likelihood $p(\,\mathbf{z}_k \,|\, \mathbf{x}_k\,)$ is obtained from the measurement model (2) and $\mathbf{z}_{1:k} \triangleq \{\mathbf{z}_1,\, \ldots,\, \mathbf{z}_k\}$. The predictive density is given by the Chapman–Kolmogorov equation $p(\,\mathbf{x}_k \,|\, \mathbf{z}_{1:k-1}\,) = \int p(\,\mathbf{x}_k \,|\, \mathbf{x}_{k-1}\,)p(\,\mathbf{x}_{k-1} \,|\, \mathbf{z}_{1:k-1}\,)\,\mathrm{d}\mathbf{x}_{k-1}$, where the transition density $p(\,\mathbf{x}_k \,|\, \mathbf{x}_{k-1}\,)$ is obtained from the system dynamics (1).

A vast majority of the well-known filters, such as EKF, UKF and GHKF, can be recovered from the Bayesian formulation under a Gaussian approximation of the joint density of the state and measurement. That is, when the density $p(\,\mathbf{x}_k, \mathbf{z}_k \,|\, \mathbf{z}_{1:k-1}\,) = p(\,\mathbf{z}_k \,|\, \mathbf{x}_k\,)p(\,\mathbf{x}_k \,|\, \mathbf{z}_{1:k-1}\,)$ is approximated by a Gaussian density of the form

$$\mathrm{N}\left(\begin{bmatrix}\mathbf{x}_k \\ \mathbf{z}_k\end{bmatrix} \,\middle|\, \begin{bmatrix}\mathbf{m}_{k|k-1}^x \\ \mathbf{m}_{k|k-1}^z\end{bmatrix},\, \begin{bmatrix}\mathbf{P}_{k|k-1}^x & \mathbf{P}_{k|k-1}^{xz} \\ \mathbf{P}_{k|k-1}^{zx} & \mathbf{P}_{k|k-1}^z\end{bmatrix}\right), \tag{3}$$

then the mean and covariance of the state posterior have analytical form, given by[1]

$$\mathbf{m}_{k|k}^x = \mathbf{m}_{k|k-1}^x + \mathbf{G}_k\Big(\mathbf{z}_k - \mathbf{m}_{k|k-1}^z\Big), \tag{4}$$
$$\mathbf{P}_{k|k}^x = \mathbf{P}_{k|k-1}^x - \mathbf{G}_k\mathbf{P}_{k|k-1}^z\mathbf{G}_k^\top, \tag{5}$$

where $\mathbf{G}_k = \mathbf{P}_{k|k-1}^{xz}\big(\mathbf{P}_{k|k-1}^z\big)^{-1}$ is the Kalman gain. The predictive moments of the state, $\mathbf{m}_{k|k-1}^x$ and $\mathbf{P}_{k|k-1}^x$, and the moments of measurements, $\mathbf{m}_{k|k-1}^z$, $\mathbf{P}_{k|k-1}^z$ and $\mathbf{P}_{k|k-1}^{xz}$, are defined as integrals of the form

$$\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})] \triangleq \int \mathbf{g}(\mathbf{x})\mathrm{N}(\,\mathbf{x} \,|\, \mathbf{m},\, \mathbf{P}\,)\,\mathrm{d}\mathbf{x}. \tag{6}$$

Table I shows which quantities have to be substituted for $\mathbf{g}(\mathbf{x})$, $\mathbf{x}$, $\mathbf{m}$ and $\mathbf{P}$ to obtain any of the above predictive moments. Since the function $\mathbf{g}$ being integrated is nonlinear in each case, these integrals cannot be typically computed analytically and some type of approximation needs to be employed. Each nonlinear filter is distinguished solely by the type of integral approximation it uses. For example, the EKF employs the first order Taylor expansion to linearise $\mathbf{g}$ in the vicinity of $\mathbf{m}$,

[1]Note that, $\mathbf{m}_{k|k}^x \triangleq \mathbb{E}_{\mathbf{x}}[\mathbf{x}_k \mid \mathbf{z}_{1:k}]$ and $\mathbf{P}_{k|k}^x \triangleq \mathbb{E}_{\mathbf{x}}\Big[(\mathbf{x}_k - \mathbf{m}_{k|k}^x)(\mathbf{x}_k - \mathbf{m}_{k|k}^x)^\top \mid \mathbf{z}_{1:k}\Big]$

Moment	$\mathbf{g}(\mathbf{x})$	$\mathbf{x}$	$\mathbf{m}$	$\mathbf{P}$
$\mathbf{m}_{k\|k-1}^x$	$\mathbf{f}(\mathbf{x}_{k-1})$	$\mathbf{x}_{k-1}$	$\mathbf{m}_{k-1\|k-1}^x$	$\mathbf{P}_{k-1\|k-1}^x$
$\mathbf{P}_{k\|k-1}^x$	$\Delta\mathbf{f}\Delta\mathbf{f}^\top$	$\mathbf{x}_{k-1}$	$\mathbf{m}_{k-1\|k-1}^x$	$\mathbf{P}_{k-1\|k-1}^x$
$\mathbf{m}_{k\|k-1}^z$	$\mathbf{h}(\mathbf{x}_k)$	$\mathbf{x}_k$	$\mathbf{m}_{k\|k-1}^x$	$\mathbf{P}_{k\|k-1}^x$
$\mathbf{P}_{k\|k-1}^z$	$\Delta\mathbf{h}\Delta\mathbf{h}^\top$	$\mathbf{x}_k$	$\mathbf{m}_{k\|k-1}^x$	$\mathbf{P}_{k\|k-1}^x$
$\mathbf{P}_{k\|k-1}^{xz}$	$\Delta\mathbf{x}\Delta\mathbf{h}^\top$	$\mathbf{x}_k$	$\mathbf{m}_{k\|k-1}^x$	$\mathbf{P}_{k\|k-1}^x$

Table I: Quantities that need to be substituted into the Gaussian integral (6) in order to obtain every predictive moment necessary to compute the moments of the state posterior. The following shorthand notation is used: $\Delta\mathbf{f} = \mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_{k|k-1}^x$, $\Delta\mathbf{h} = \mathbf{h}(\mathbf{x}_k) - \mathbf{m}_{k|k-1}^z$, $\Delta\mathbf{x} = \mathbf{x}_k - \mathbf{m}_{k|k-1}^x$.

which in turn facilitates analytic tractability of the moment integrals. On the other hand, the sigma-point filters, such as the UKF and the GHKF, leverage numerical quadrature for approximation of the integral. Since quadratures are typically designed to be used with standard Gaussian, the integrals of the form (6) need to be converted by employing a stochastic decoupling substitution $\mathbf{x}^{(n)} = \mathbf{m} + \mathbf{L}\boldsymbol{\xi}^{(n)}$, which leads to an approximation

$$\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})] \approx \sum_{n=1}^{N} w_n\mathbf{g}(\mathbf{m} + \mathbf{L}\boldsymbol{\xi}^{(n)}) = \sum_{n=1}^{N} w_n\tilde{\mathbf{g}}(\boldsymbol{\xi}^{(n)}), \tag{7}$$

where $\boldsymbol{\xi}^{(n)}$ denotes the n-th unit sigma-point, $w_n \in \mathbb{R}$ is the n-th weight, N is the total number of sigma-points, $\mathbf{L}$ is a matrix factor such that $\mathbf{P} = \mathbf{L}\mathbf{L}^\top$ and $\tilde{\mathbf{g}}(\boldsymbol{\xi}) \triangleq \mathbf{g}(\mathbf{m} + \mathbf{L}\boldsymbol{\xi})$. Note that various quadrature rules are distinguished by the different weights and sigma-points they prescribe to satisfy various optimality criteria.

III. Sigma-Point Moment Transforms

From the above exposition, it is apparent that the central issue in filtering is the design of the so-called moment transformations, which generate approximations to the moments of a random variable under a nonlinear transformation.

Let $\mathbf{x} \in \mathbb{R}^D$ be an input Gaussian random variable and $\mathbf{y} \in \mathbb{R}^E$ an output random variable defined by

$$\mathbf{y} = \mathbf{g}(\mathbf{x}), \quad \mathbf{x} \sim \mathrm{N}(\mathbf{m},\, \mathbf{P}). \tag{8}$$

If the transformation $\mathbf{g}$ is nonlinear, the joint density $p(\mathbf{x},\, \mathbf{y})$ will be non-Gaussian in general. However, there are many applied situations where $\mathbf{g}$ is approximately linear in the region where probability mass is concentrated. In such situations the principal error term in the moment transform is numerical quadrature error. This error is the focus of our present work and, therefore, in what follows we proceed under the assumption that the Gaussian approximation

$$\mathrm{N}\left(\begin{bmatrix}\mathbf{x} \\ \mathbf{y}\end{bmatrix} \,\middle|\, \begin{bmatrix}\mathbf{m} \\ \boldsymbol{\mu}\end{bmatrix},\, \begin{bmatrix}\mathbf{P} & \mathbf{C} \\ \mathbf{C}^\top & \boldsymbol{\Pi}\end{bmatrix}\right) \tag{9}$$

of $p(\mathbf{x},\, \mathbf{y})$ can be justified. In this setting the moment transformation then reduces to computing the output mean $\boldsymbol{\mu}$, covariance $\boldsymbol{\Pi}$ and cross-covariance $\mathbf{C}$ as accurately as possible, when supplied with the input moments, $\mathbf{m}$ and $\mathbf{P}$. This is a specific instance of *uncertainty propagation* [19].

In this article we focus on the sigma-point approximations, exemplified by eq. (7), to the moment integrals in Table I. The well-known classical approximations, such as the Gauss–Hermite, the spherical-radial and the unscented transform, are conventionally written in the form

$$\boldsymbol{\mu} \approx \hat{\boldsymbol{\mu}} = \sum_{n=1}^{N} w_n \tilde{\mathbf{g}}(\boldsymbol{\xi}^{(n)}), \tag{10}$$

$$\boldsymbol{\Pi} \approx \hat{\boldsymbol{\Pi}} = \sum_{n=1}^{N} w_n \big(\tilde{\mathbf{g}}(\boldsymbol{\xi}^{(n)}) - \hat{\boldsymbol{\mu}}\big)\big(\tilde{\mathbf{g}}(\boldsymbol{\xi}^{(n)}) - \hat{\boldsymbol{\mu}}\big)^\top, \tag{11}$$

$$\mathbf{C} \approx \hat{\mathbf{C}} = \mathbf{L} \sum_{n=1}^{N} w_n \boldsymbol{\xi}^{(n)} \big(\tilde{\mathbf{g}}(\boldsymbol{\xi}^{(n)}) - \hat{\boldsymbol{\mu}}\big)^\top, \tag{12}$$

which, under the assumption that $\sum w_n = 1$ and $\sum w_n \boldsymbol{\xi}^{(n)} = 0$, we will prefer to write using the matrix notation as

$$\hat{\boldsymbol{\mu}} = \mathbf{Y}^\top \mathbf{w}, \tag{13a}$$

$$\hat{\boldsymbol{\Pi}} = \mathbf{Y}^\top \mathbf{W} \mathbf{Y} - \hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\top, \tag{13b}$$

$$\hat{\mathbf{C}} = \mathbf{L}\boldsymbol{\Xi}\mathbf{W}_c\mathbf{Y}, \tag{13c}$$

where $\boldsymbol{\Xi} = \begin{bmatrix}\boldsymbol{\xi}^{(1)} & \dots & \boldsymbol{\xi}^{(N)}\end{bmatrix}$ and the matrix of integrand evaluations is given by $[\mathbf{Y}]_{ne} \triangleq \tilde{g}_e(\boldsymbol{\xi}^{(n)})$, where e indexes outputs of $\tilde{\mathbf{g}}$ and $[\,\cdot\,]_{ne}$ denotes the matrix element at position (n, e). The vector $\mathbf{w}$ contains the weights and $\mathbf{W} = \mathbf{W}_c = \mathrm{diag}(\,\mathbf{w}\,)$ for any classical sigma-point moment transform. Each moment transform uses a different set of sigma-points and weights.

A. Unscented Transform

The unscented transform (UT) of D-dimensional input uses $N = 2D + 1$ sigma-points, which exploit symmetry of the Gaussian distribution, given, for $d = 1, \dots, D$, by

$$\boldsymbol{\xi}^{(0)} = \mathbf{0}, \quad \boldsymbol{\xi}^{(d)} = \sqrt{c}\,\mathbf{e}_d, \quad \boldsymbol{\xi}^{(D+d)} = -\sqrt{c}\,\mathbf{e}_{D+d}, \tag{14}$$

where $\mathbf{e}_d$ is the standard unit vector and $c = D + \kappa$ for a scaling parameter κ. The weights are defined as $w_0 = \frac{\kappa}{c}$, $w_d = w_{D+d} = \frac{1}{2c}$. This selection of sigma-points and weights yields a quadrature rule that integrates exactly all polynomials of (total) degree at most three; the derivation is essentially contained in the proof of Theorem 1. The *spherical-radial rule*, which is used in the cubature Kalman filter (CKF) [20], is equivalent to the UT with $\kappa = 0$; it therefore lacks the central sigma-point.

B. Gauss–Hermite Rule

From non-singularity of the Vandermonde matrix $[\mathbf{V}]_{nm} = x_n^{m-1}$ for any distinct sigma-points $x_1, \dots, x_p \in \mathbb{R}$ it follows that there are unique weights such that $\sum_{n=1}^{p} w_n x_n^m = \int x^m \mathrm{N}(x \mid 0, 1)\,\mathrm{d}x$ for every $m \leq p - 1$ (i.e., the rule has a degree of exactness $p - 1$). However, degree of exactness $2p-1$ can be achieved with p sigma-points if these are selected to be the roots of the p-th degree Hermite polynomial H_p. The weights are then given by $w_n = \frac{p!}{p^2 \mathrm{H}_{p-1}(\xi^{(n)})^2}$. This is the Gauss–Hermite (GH) rule [21]–[24]. In multivariate versions, the sigma-points are formed as Cartesian products of the aforementioned one-dimensional points and the weights are products of w_n. The multivariate GH rule exactly integrates functions in the space

$$\Pi_{2p-1}^{\max} \triangleq \mathrm{span}\left\{\mathbf{x}^{\boldsymbol{\alpha}} :\ \boldsymbol{\alpha} \in \mathbb{N}_0^D,\ \max_{d=1,\dots,D} \alpha_d \leq 2p-1\right\}, \tag{15}$$

where $\mathbf{x}^{\boldsymbol{\alpha}} = \prod_{d=1}^{D} x_d^{\alpha_d}$ denotes multivariate monomial. Because of the Cartesian product design, the number of points, $N = p^D$, in the GH rule grows exponentially with dimension, which makes it practically unattractive for $D > 5$ [22]. The problem can be partially mitigated by using sparse grids [25].

IV. Bayesian Quadrature

This section reviews the underlying philosophy of the Bayesian quadrature as an alternative perspective on numerical integration and describes the Bayes–Sard quadrature as a necessary stepping stone on the way to building the Bayes–Sard moment transform proposed in Section V. A general formulation of the BQ is presented for integrals

$$\mathbb{E}_{\mathbf{x}}\big[g^\dagger(\mathbf{x})\big] = \int g^\dagger(\mathbf{x}) p(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{16}$$

with arbitrary density function p. Vector-valued integrands are discussed in Section IV-B. The moment transform proposed in Section V then specialises to the case $p(\mathbf{x}) = \mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I})$. Throughout this section, the true integrand will be denoted by $g^\dagger$ to distinguish it from the stochastic model of the integrand.

From eq. (7) it is clear that the quadrature approximation of the integral (6) is based on limited knowledge about the behaviour of the integrand, because it only relies on finitely many evaluations. The design of classical quadrature rules typically involves formation of polynomial interpolant passing through the observed function values, which is then integrated instead of the intractable integrand. The polynomial interpolation of the integrand consequently implies that the classical rules are only able to integrate polynomial integrands exactly. Another downside of the classical rules is that they are unable to account for the functional uncertainty (interpolation error), which occurs when the integrand is not a polynomial.

The Bayesian approach to quadrature [11], [26], [27] aims to address these limitations by treating the numerical approximation of intractable integrals as a problem of Bayesian statistical inference, where a prior for the integrand is specified by a stochastic process model $g(\mathbf{x})$ with user-defined mean function $m(\mathbf{x}) = \mathbb{E}_g[g(\mathbf{x})]$ and covariance (or kernel) function $k(\mathbf{x}, \mathbf{x}') = \mathbb{C}_g[g(\mathbf{x}), g(\mathbf{x}')]$, where $\mathbf{x}'$ denotes the second argument (not a transpose). The dataset $\mathcal{D} = \big\{(\mathbf{x}^{(n)}, g^\dagger(\mathbf{x}^{(n)}))\big\}_{n=1}^{N}$ comprises evaluations of the integrand $g^\dagger(\mathbf{x}^{(n)})$ at pre-defined points $\mathbf{x}^{(n)}$. Conditioning on $\mathcal{D}$ leads to a posterior stochastic process, with mean $m_{\mathcal{D}}(\mathbf{x}) = \mathbb{E}_{g|\mathcal{D}}[g(\mathbf{x})]$ and covariance $k_{\mathcal{D}}(\mathbf{x}, \mathbf{x}') = \mathbb{C}_{g|\mathcal{D}}[g(\mathbf{x}), g(\mathbf{x}')]$, which in turn induces a posterior marginal distribution on the value of the integral $\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]$, with the first two moments given by [28]

$$\mathbb{E}_{g|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbb{E}_{\mathbf{x}}\big[\mathbb{E}_{g|\mathcal{D}}[g(\mathbf{x})]\big], \tag{17}$$

$$\mathbb{V}_{g|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbb{E}_{\mathbf{x},\mathbf{x}'}\big[\mathbb{C}_{g|\mathcal{D}}[g(\mathbf{x}), g(\mathbf{x}')]\big]. \tag{18}$$

The mean is a convenient point estimate while the full posterior serves as a probabilistic model of the integration error. The most common stochastic process model of the integrand is a Gaussian process (GP), which has been studied extensively [11], [29].

A. Bayes–Sard Gaussian Process Model

Let π be a linear function space spanned by $Q \leq N$ functions $\phi_1, \ldots, \phi_Q \colon \mathbb{R}^D \to \mathbb{R}$. Modeling of the scalar integrand $g^\dagger : \mathbb{R}^D \to \mathbb{R}$ in Bayes–Sard quadrature (BSQ) begins by considering a hierarchical GP prior given by

$$\boldsymbol{\gamma} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_\pi), \quad \boldsymbol{\gamma} \in \mathbb{R}^Q \tag{19}$$

$$m(\mathbf{x}) = \sum_{q=1}^{Q} \gamma_q \phi_q(\mathbf{x}), \tag{20}$$

$$g(\mathbf{x}) \sim \mathrm{GP}(m(\mathbf{x}),\, k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})), \tag{21}$$

where the prior mean function $m(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}$ is composed of basis functions $\phi_q(\mathbf{x})$ of Q-dimensional linear space π and the prior covariance function (kernel) $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ can be any symmetric positive-definite function parametrized by the vector $\boldsymbol{\theta}$ (see section IV-E for concrete example). The dependence on $\boldsymbol{\theta}$ will be tacitly assumed and explicitly denoted only when required. Discussion about the particular choice of the kernel and its effects is postponed to Section IV-E. The above model differs from the one often used in GP-based Bayesian quadrature, in that the prior mean function is non-zero and its coefficients are random.

The next phase in modelling is to consider a flat prior limit on the mean function coefficients, such that $\boldsymbol{\Sigma}_\pi \to \infty$ [30, Chapter 4]. In order for the GP posterior to be well-defined, the set $\mathcal{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ of sigma-points must meet the following condition of π-unisolvency, which is related to existence of interpolants formed out of linear combinations of $\phi_1, \ldots, \phi_q$.

Definition 1 (π-unisolvency)**.** Let π be a Q-dimensional linear space spanned by $\{\phi_1, \ldots, \phi_q\}$. A point set $\mathcal{X}$ is said to be π-unisolvent if and only if the $N \times Q$ alternant matrix $\left[\boldsymbol{\Phi}\right]_{qn} \triangleq \phi_q(\mathbf{x}^{(n)})$ is of full-rank.

We further restrict the model to the case when $N = Q$, which means the alternant matrix $\boldsymbol{\Phi}$ is square and, due to π-unisolvency of $\mathcal{X}$, invertible.

With all the assumptions laid out, the final step is to condition the GP on the set of sigma-points $\mathcal{X}$ and the corresponding integrand evaluations, to arrive at the posterior moments of the Bayes–Sard GP model given by [15] as

$$\mathbb{E}_{g|\mathcal{D}}[g(\mathbf{x})] = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Phi}^{-1} \mathbf{y}, \tag{22}$$

$$\begin{aligned}\mathbb{C}_{g|\mathcal{D}}[g(\mathbf{x}), g(\mathbf{x}')] = {} & k(\mathbf{x}, \mathbf{x}') - 2\mathbf{k}(\mathbf{x})^\top \boldsymbol{\Phi}^{-\top} \boldsymbol{\phi}(\mathbf{x}') \\ & + \boldsymbol{\phi}(\mathbf{x})^\top \left[\boldsymbol{\Phi}^\top \mathbf{K}^{-1} \boldsymbol{\Phi}\right]^{-1} \boldsymbol{\phi}(\mathbf{x}'),\end{aligned} \tag{23}$$

where $\left[\mathbf{k}(\mathbf{x})\right]_n \triangleq k(\mathbf{x}, \mathbf{x}^{(n)})$, $\left[\boldsymbol{\phi}(\mathbf{x})\right]_q \triangleq \phi_q(\mathbf{x})$ and $\left[\mathbf{y}\right]_n \triangleq g^\dagger(\mathbf{x}^{(n)})$, where $\left[\,\cdot\,\right]_n$ denotes the n-th element of the given vector. Note that the posterior mean now only depends on the choice of the function space π and the kernel affects only the posterior covariance. It is worth pointing out that all sigma-point sets in the established classical filters are π-unisolvent.

B. Vector-Valued Integrands

Until now, we have only considered scalar-valued integrands. The model specified by eqs. (22) and (23) can be straightforwardly extended to vector-valued integrands $\mathbf{g}^\dagger : \mathbb{R}^D \to \mathbb{R}^E$ that comply with the specification of the moment transformation problem in eq. (8). Noticing that we can decompose the integrand as $\mathbf{g}^\dagger(\mathbf{x}) = \begin{bmatrix} g_1^\dagger(\mathbf{x}) & \ldots & g_E^\dagger(\mathbf{x}) \end{bmatrix}^\top$, the simplest solution is to use eqs. (22) and (23) to model each coordinate function independently, either using a common kernel parameter for all outputs, which is accomplished by

$$g_e(\mathbf{x}) \mid \mathcal{D} \sim \mathrm{GP}(m_{\mathcal{D}}(\mathbf{x}),\, k_{\mathcal{D}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})), \tag{24}$$

or using a different kernel parameter values for each output, so that

$$g_e(\mathbf{x}) \mid \mathcal{D} \sim \mathrm{GP}(m_{\mathcal{D}}(\mathbf{x}),\, k_{\mathcal{D}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_e)) \tag{25}$$

for all $e = 1, \ldots, E$. In both cases, the GP posterior mean function is given as

$$\mathbf{m}_{\mathcal{D}}(\mathbf{x}) \triangleq \mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x})] = \mathbf{Y}^\top \boldsymbol{\Phi}^{-1} \boldsymbol{\phi}(\mathbf{x}), \tag{26}$$

where $\left[\mathbf{Y}\right]_{ne} = g_e^\dagger(\mathbf{x}^{(n)})$. For the single-parameter model (24) the posterior covariance becomes

$$\mathbf{K}_{\mathcal{D}}(\mathbf{x}, \mathbf{x}') \triangleq \mathbb{C}_{\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x}), \mathbf{g}(\mathbf{x}')] = k_{\mathcal{D}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})\, \mathbf{I}_E \tag{27}$$

and for the multi-parameter model (25), we get

$$\mathbf{K}_{\mathcal{D}}(\mathbf{x}, \mathbf{x}') = \mathrm{diag}\big(\begin{bmatrix} k_{\mathcal{D}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_1) & \ldots & k_{\mathcal{D}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_E) \end{bmatrix}\big). \tag{28}$$

Both of these modelling choices assume that the outputs are conditionally independent given the inputs. Alternatively, the use of multi-output Gaussian processes [31], [32] would make it possible to model correlations between coordinate functions and use coordinate-dependent sigma-points at the expense of increased computational cost.

C. Bayes–Sard Quadrature

The advantage of using a GP for modelling the integrand is that as it gets transformed by the integral, which is a linear operator, the resulting distribution over the value of the integral is also Gaussian. The Bayes–Sard quadrature [15], [27], [33] enables enforcing exactness conditions of the form

$$\mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})]] = \int \mathbf{g}^\dagger(\mathbf{x}) p(\mathbf{x})\, \mathrm{d}\mathbf{x}$$

for all functions $\mathbf{g}^\dagger \colon \mathbb{R}^D \to \mathbb{R}^E$ such that $g_e^\dagger \in \pi$ for each $e = 1, \ldots, E$. As shown in Section IV-D, the classical quadrature methods can be replicated by judicious choice of the function space π. The posterior integral mean and variance under the Bayes–Sard quadrature are straightforwardly derived by plugging the Bayes–Sard GP model moments from eqs. (26) and (28) into the general BQ expressions in eqs. (17) and (18).

For the mean of the posterior distribution of the integral, we have

$$\mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})]] = \mathbb{E}_{\mathbf{x}}[\mathbf{m}_{\mathcal{D}}(\mathbf{x})] = \mathbf{Y}^\top \boldsymbol{\Phi}^{-1} \bar{\boldsymbol{\phi}}, \tag{29}$$

where $\left[\bar{\boldsymbol{\phi}}\right]_q = \mathbb{E}_{\mathbf{x}}[\phi_q(\mathbf{x})] = \int \phi_q(\mathbf{x}) p(\mathbf{x}) \,\mathrm{d}\mathbf{x}$. Recognizing that the vector of quadrature weights is $\mathbf{w} = \boldsymbol{\Phi}^{-1}\bar{\boldsymbol{\phi}}$, we see that the posterior mean of the integral

$$\mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})]] = \mathbf{Y}^\top \mathbf{w} = \sum_{n=1}^{N} w_n \mathbf{g}^\dagger(\mathbf{x}^{(n)}), \tag{30}$$

takes on the form of weighted sum from eq. (7). The integral covariance becomes

$$\mathbb{V}_{\mathbf{g}|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})]] = \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\mathbf{K}_{\mathcal{D}}(\mathbf{x},\mathbf{x}')] = \mathrm{diag}\big(\begin{bmatrix} \bar{k}^1_{\mathcal{D}} & \dots & \bar{k}^E_{\mathcal{D}} \end{bmatrix} \big) \tag{31}$$

where $\bar{k}^e_{\mathcal{D}} \triangleq \mathbb{E}_{\mathbf{x},\mathbf{x}'}[k_{\mathcal{D}}(\mathbf{x},\mathbf{x}';\boldsymbol{\theta}_e)]$ and

$$\bar{k}^e_{\mathcal{D}} = \bar{k} - 2\bar{\mathbf{k}}^\top \boldsymbol{\Phi}^{-\top} \bar{\boldsymbol{\phi}} + \bar{\boldsymbol{\phi}}^\top \left[\boldsymbol{\Phi}^\top \mathbf{K}^{-1} \boldsymbol{\Phi}\right]^{-1} \bar{\boldsymbol{\phi}}. \tag{32}$$

Since the single-parameter model in eq. (27) is a special case of eq. (28), the posterior integral variance under this model would be a trivial modification of eq. (31).

D. Relationship to Classical Sigma-Point Rules

As stated in the previous section, careful selection of π (via the basis functions ϕ_q) allows for recovery of many well-known classical quadrature rules used in nonlinear filtering. Below, we show that the unscented transform and the Gauss–Hermite rule are special cases of the BSQ whenever the space π is spanned by suitably selected polynomial basis. Similar results can be proved for many other sigma-point rules. Note that the BSQ reports a non-zero integral variance even for $\mathbf{g}^\dagger$ whose coordinate functions are in π (and hence integrated exactly). This behaviour is desirable because, given only a finite set of function values, one can never tell with certainty the true nature of the integrand.

Theorem 1. *Consider the standard Gaussian distribution,* $p(\mathbf{x}) = \mathrm{N}(\mathbf{x} \,|\, \mathbf{0}, \mathbf{I})$. *Select the* $2D+1$ *dimensional function space*

$$\pi = \mathrm{span}\left\{1,\, x_1,\, \dots,\, x_D,\, x_1^2,\, \dots,\, x_D^2\right\} \tag{33}$$

and the $N = 2D+1$ *unscented transform points* (14)*. Then, the Bayes–Sard weights* $\mathbf{w} = \boldsymbol{\Phi}^{-1}\bar{\boldsymbol{\phi}}$ *that determine the posterior mean* (29) *coincide with the unscented transform weights.*

Proof: Because $\dim(\pi) = N$, the Bayes–Sard weights $\mathbf{w}$ solve the linear system $\boldsymbol{\Phi}\mathbf{w} = \bar{\boldsymbol{\phi}}$. That is, they are the unique weights such that

$$\sum_{n=0}^{2D} w_n v(\boldsymbol{\xi}^{(n)}) = \int v(\mathbf{x}) \mathrm{N}(\mathbf{x} \,|\, \mathbf{0}, \mathbf{I}) \,\mathrm{d}\mathbf{x} \tag{34}$$

for every polynomial $v \in \pi$. In the following, let $d = 1, \dots, D$. We have $\int \mathrm{N}(\mathbf{x} \,|\, \mathbf{0}, \mathbf{I}) \,\mathrm{d}\mathbf{x} = 1$, $\int x_d \mathrm{N}(\mathbf{x} \,|\, \mathbf{0}, \mathbf{I}) \,\mathrm{d}\mathbf{x} = 0$ and $\int x_d^2 \mathrm{N}(\mathbf{x} \,|\, \mathbf{0}, \mathbf{I}) \,\mathrm{d}\mathbf{x} = 1$. Consequently, eq. (34) is equivalent to $\sum_{n=0}^{2D} w_n = 1$, $\sum_{n=0}^{2D} w_n x_{n,d} = 0$ and $\sum_{n=0}^{2D} w_n x_{n,d}^2 = 1$. Because $\boldsymbol{\xi}_d = -\boldsymbol{\xi}_{D+d}$, the second of these equations implies that $w_d = w_{D+d}$, while the third one yields $w_d = w_{D+d} = \frac{1}{2c}$. Furthermore, $w_0 = \frac{\kappa}{c}$ due to the weights summing up to one. We have thus solved the BSQ weights $\mathbf{w} = \boldsymbol{\Phi}^{-1}\bar{\boldsymbol{\phi}}$ and see that they are precisely UT weights in section III-A. ■

Theorem 2. *Consider the standard Gaussian distribution,* $p(\mathbf{x}) = \mathrm{N}(\mathbf{x} \,|\, \mathbf{0}, \mathbf{I})$, *and let* $p \geq 1$. *Select the* p^D *dimensional function space* $\pi = \Pi^{max}_{p-1} \triangleq \mathrm{span}\left\{\mathbf{x}^{\boldsymbol{\alpha}} :\, \boldsymbol{\alpha} \in \mathbb{N}_0^D,\, \max_{d=1,\dots,D} \alpha_d \leq p-1\right\}$, *and the points that constitute the Cartesian product of the roots of the* p*-th degree Hermite polynomial. Then, the Bayes–Sard weights* $\mathbf{w} = \boldsymbol{\Phi}^{-1}\bar{\boldsymbol{\phi}}$ *that determine the posterior mean* (29) *coincide with the classical Gauss–Hermite weights from Section III-B.*

Proof: Since the Bayes–Sard weights yield, by their definition, a quadrature rule exact for functions in π and it is known that, given the Gauss–Hermite points, the Gauss–Hermite weights are the unique weights that determine a quadrature rule exact for this very same function space (see Section III-B), the result follows. ■

E. Choice of Kernel

As already noted, the posterior mean for the integral produced by the BSQ depends only on π and the kernel controls the posterior variance of the integral. The reasonableness of the BSQ output depends on the reasonableness of the assumption that $g^\dagger$ is "well modelled" by the GP specified by the kernel k. Consequently, selection of the kernel is important in order to ensure that the integral variance is meaningful in modelling the integration error. At the same time, the functional form of the kernel is constrained by the requirement in BSQ to analytically compute the integral of the kernel. To facilitate analytic tractability of the Bayes–Sard moment transform, introduced next, we use the radial basis function (RBF) kernel

$$k(\mathbf{x},\mathbf{x}') = \alpha^2 \prod_{d=1}^{D} \exp\left(-\frac{(x_d - x'_d)^2}{2\ell_d^2}\right) \tag{35}$$

throughout the remainder. The parameters $\boldsymbol{\theta}$ of this kernel consist of the scale parameter $\alpha > 0$ and dimension-wise lengthscale parameters $\ell_1, \dots, \ell_D > 0$. A particular modelling assumption associated with this kernel is that the integrand is infinitely differentiable. If this is not the case (i.e., there is model misspecification) the proposed method still works but the uncertainty quantification for the integral may be rendered less meaningful. For certain classes of kernels it has been shown that convergence rates to the true integral as $N \to \infty$ are not much affected by model misspecification [34].

V. Bayes–Sard Moment Transform

The simplest way to design a moment transform is to use the BSQ directly for approximation of the moment integrals in eqs. (13a) to (13c). However, this design does not reflect integral uncertainty, which is the key advantage of Bayesian quadrature, not to mention the fact that we would only obtain the classical rules as a result. To resolve this issue, we employ

the same general conceptual framework used in the design of the GPQ moment transform in [14], which can account for the variance of the mean integral (13a).

A. Incorporating Integration Error

First, it is important to realize that the output variable $\mathbf{y}$ is now subject to an additional source of uncertainty in $\mathbf{g}$ introduced by the model. The key idea is to account for all sources of uncertainty in the computed moments, which can be achieved with the following

$$\boldsymbol{\mu} = \mathbb{E}_{\mathbf{x}}\big[\mathbf{g}^{\dagger}(\mathbf{x})\big] \quad \approx \quad \hat{\boldsymbol{\mu}} = \mathbb{E}_{\mathbf{x},\,\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x})] \tag{36}$$

$$\boldsymbol{\Pi} = \mathbb{C}_{\mathbf{x}}\big[\mathbf{g}^{\dagger}(\mathbf{x}), \mathbf{g}^{\dagger}(\mathbf{x})\big] \quad \approx \quad \hat{\boldsymbol{\Pi}} = \mathbb{C}_{\mathbf{x},\,\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x}),\, \mathbf{g}(\mathbf{x})] \tag{37}$$

$$\mathbf{C} = \mathbb{C}_{\mathbf{x}}\big[\mathbf{x}, \mathbf{g}^{\dagger}(\mathbf{x})\big] \quad \approx \quad \hat{\mathbf{C}} = \mathbb{C}_{\mathbf{x},\,\mathbf{g}|\mathcal{D}}[\mathbf{x},\, \mathbf{g}(\mathbf{x})] \tag{38}$$

Using the law of total expectation and covariance, the approximate moments of the output can be written as

$$\hat{\boldsymbol{\mu}} = \mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})]] = \mathbb{E}_{\mathbf{x}}\big[\mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x})]\big], \tag{39}$$

$$\hat{\boldsymbol{\Pi}} = \mathbb{C}_{\mathbf{g}|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})]] + \mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbb{C}_{\mathbf{x}}[\mathbf{g}(\mathbf{x}), \mathbf{g}(\mathbf{x})]], \tag{40}$$

$$= \mathbb{C}_{\mathbf{x}}\big[\mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x})]\big] + \mathbb{E}_{\mathbf{x}}\big[\mathbb{C}_{\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x}), \mathbf{g}(\mathbf{x})]\big], \tag{41}$$

$$\hat{\mathbf{C}} = \mathbb{E}_{\mathbf{x}}\big[\mathbf{x}\,\mathbb{E}_{\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x})]\big] - \mathbb{E}_{\mathbf{x}}[\mathbf{x}]\mathbb{E}_{\mathbf{g}|\mathcal{D},\mathbf{x}}[\mathbf{g}(\mathbf{x})]. \tag{42}$$

The first equality exposes the fact that integral mean is obtained by integrating the mean function of the integrand model. The way the integral uncertainty is incorporated into the output covariance is revealed by eq. (40). Note that since the model of the integrand has conditionally independent outputs, the covariance of the integral, $\mathbb{C}_{\mathbf{g}|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})]]$, and the model covariance, $\mathbb{C}_{\mathbf{g}|\mathcal{D}}[\mathbf{g}(\mathbf{x}), \mathbf{g}(\mathbf{x})]$, are diagonal matrices. When either of the covariances approaches zero, eqs. (39) to (42) approach their true values. From now on, we will work with the output covariance in the form (41) because it is easier to analyse and implement.

B. Derivation of Transformed Moments

In the following derivations, explicit conditioning on $\mathcal{D}$ in the expectations is omitted to reduce notational clutter. We also assume that the stochastic decoupling substitution has taken place in the integrals, so that $\tilde{\mathbf{g}}(\boldsymbol{\xi}) = \mathbf{g}(\mathbf{m} + \mathbf{L}\boldsymbol{\xi})$.

Taking the expression for the mean function of the model in eq. (26) and plugging it into eq. (39), the output mean of the Bayes–Sard quadrature moment transform (BSQMT) becomes

$$\hat{\boldsymbol{\mu}} = \mathbb{E}_{\boldsymbol{\xi}}[\mathbb{E}_{\mathbf{g}}[\tilde{\mathbf{g}}(\boldsymbol{\xi})]] = \mathbf{Y}^{\top}\boldsymbol{\Phi}^{-\top}\mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{\phi}(\boldsymbol{\xi})] = \mathbf{Y}^{\top}\mathbf{w}, \tag{43}$$

where $\mathbf{w} = \boldsymbol{\Phi}^{-\top}\mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{\phi}(\boldsymbol{\xi})]$ are the mean weights. The output covariance becomes

$$\hat{\boldsymbol{\Pi}} = \mathbb{E}_{\boldsymbol{\xi}}\big[\mathbb{E}_{\mathbf{g}}[\tilde{\mathbf{g}}(\boldsymbol{\xi})]\,\mathbb{E}_{\mathbf{g}}[\tilde{\mathbf{g}}(\boldsymbol{\xi})]^{\top}\big] - \hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^{\top} + \mathbb{E}_{\boldsymbol{\xi}}[\mathbb{C}_{\mathbf{g}}[\tilde{\mathbf{g}}(\boldsymbol{\xi}), \tilde{\mathbf{g}}(\boldsymbol{\xi})]]$$
$$= \mathbf{Y}^{\top}\mathbf{W}\mathbf{Y} - \hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^{\top} + \bar{\sigma}^2\mathbf{I}_E \tag{44}$$

where the expected model variance is

$$\bar{\sigma}^2 = \mathbb{E}_{\boldsymbol{\xi}}[k(\boldsymbol{\xi}, \boldsymbol{\xi})] - \operatorname{tr}\big[\mathbf{D}^{\top}\boldsymbol{\Phi}^{-\top} + \mathbf{D}\boldsymbol{\Phi}^{-1} - \mathbf{W}\mathbf{K}\big]. \tag{45}$$

Here $\mathbf{D} = \mathbb{E}_{\boldsymbol{\xi}}\big[\mathbf{k}(\boldsymbol{\xi})\boldsymbol{\phi}(\boldsymbol{\xi})^{\top}\big]$ and the covariance weights are $\mathbf{W} = \boldsymbol{\Phi}^{-\top}\mathbb{E}_{\boldsymbol{\xi}}\big[\boldsymbol{\phi}(\boldsymbol{\xi})\boldsymbol{\phi}(\boldsymbol{\xi})^{\top}\big]\boldsymbol{\Phi}^{-1}$. Finally, the covariance between the input and output becomes

$$\hat{\mathbf{C}} = \mathbb{E}_{\boldsymbol{\xi}}[(\mathbf{m} + \mathbf{L}\boldsymbol{\xi})\,\mathbb{E}_{\mathbf{g}}[\tilde{\mathbf{g}}(\boldsymbol{\xi})]] - \mathbb{E}_{\boldsymbol{\xi}}[\mathbf{m} + \mathbf{L}\boldsymbol{\xi}]\,\mathbb{E}_{\mathbf{g},\boldsymbol{\xi}}[\tilde{\mathbf{g}}(\boldsymbol{\xi})]$$
$$= \mathbf{L}\mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{\xi}\,\boldsymbol{\phi}(\boldsymbol{\xi})]\boldsymbol{\Phi}^{-1}\mathbf{Y} = \mathbf{L}\mathbf{W}_c\mathbf{Y} \tag{46}$$

where the cross-covariance weights are $\mathbf{W}_c = \mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{\xi}\,\boldsymbol{\phi}(\boldsymbol{\xi})]\boldsymbol{\Phi}^{-1}$.

It has now become evident that the output moments depend on the expectations of the basis functions. In Section IV, we have shown that the classical moment transforms can be recovered when the basis functions are multivariate polynomials. When this basis and the RBF kernel eq. (35) are used, the expectations above are available in closed form. The complete algorithm of the Bayes–Sard moment transform is summarized in Alg. 1.

Algorithm 1: Bayes-Sard quadrature moment transform

Input: The mean $\mathbf{m}$ and the covariance $\mathbf{P}$ of the input variable $\mathbf{x}$, the integrand $\mathbf{g}(\mathbf{x})$, the matrix of unit sigma-points $\boldsymbol{\Xi}$ and the kernel parameters $\boldsymbol{\theta}$.

Output: Approximate mean $\hat{\boldsymbol{\mu}}$ and covariance $\hat{\boldsymbol{\Pi}}$ of the output variable $\mathbf{y} = \mathbf{g}(\mathbf{x})$, and approximate input-output covariance $\hat{\mathbf{C}}$.

Function `BSQMT(`$\mathbf{g}(\mathbf{x})$, $\mathbf{m}$, $\mathbf{P}$, $\boldsymbol{\Xi}$, $\boldsymbol{\theta}$`)`

```
    // form sigma-points
1   L ← MatrixFactor(P)
2   X ← m + LΞ
3   k̄ ← E_ξ[k(ξ, ξ; θ)]
4   φ̄ ← E_ξ[φ(ξ)]
5   A ← E_ξ[φ(ξ)φ(ξ)^⊤]
6   B ← E_ξ[ξφ(ξ)^⊤]
7   for n ← 1 to N do
8       [Y]_{n*} ← g(x^(n))
9       [Φ]_{n*} ← [φ_1(ξ^(n)) ... φ_N(ξ^(n))]
10      [D]_{n*} ← E_ξ[k(ξ, ξ^(n); θ)φ(ξ)^⊤]
11      for m ← 1 to N do
12          [K]_{nm} ← k(ξ^(n), ξ^(m); θ)
13      end
14  end
    // compute BSQ weights
15  w ← Φ^{-⊤} φ̄
16  W ← Φ^{-⊤} A Φ^{-1}
17  W_c ← B Φ^{-1}
    // compute transformed moments
    // using eqs. (43) to (46)
18  return μ̂, Π̂, Ĉ
```

Theorem 3. *The BSQ output covariance $\hat{\boldsymbol{\Pi}}$ is positive semi-definite.*

Proof: Using the expression for the BSQ mean weights from Alg. 1, we can write the output covariance as $\hat{\mathbf{\Pi}} = \mathbf{Y}^\top \mathbf{\Phi}^{-\top}(\mathbf{A} - \bar{\boldsymbol{\phi}}\bar{\boldsymbol{\phi}}^\top)\mathbf{\Phi}^{-1}\mathbf{Y} + \bar{\sigma}^2\mathbf{I}_E$. Define $\mathbf{Z} = \mathbf{\Phi}^{-1}\mathbf{Y}$ and $\tilde{\mathbf{A}} = \mathbf{A} - \bar{\boldsymbol{\phi}}\bar{\boldsymbol{\phi}}^\top$, then $\mathbf{\Pi} = \mathbf{Z}^\top\tilde{\mathbf{A}}\mathbf{Z} + \bar{\sigma}^2\mathbf{I}_E$. We recognize that $\tilde{\mathbf{A}} = \mathbb{V}[\boldsymbol{\phi}(\boldsymbol{\xi})] = \mathbb{E}\big[\boldsymbol{\phi}(\boldsymbol{\xi})\boldsymbol{\phi}(\boldsymbol{\xi})^\top\big] - \mathbb{E}[\boldsymbol{\phi}(\boldsymbol{\xi})]\mathbb{E}[\boldsymbol{\phi}(\boldsymbol{\xi})]^\top \succeq 0$, which follows from the properties of covariance matrices. This implies that $\mathbf{Z}^\top\tilde{\mathbf{A}}\mathbf{Z} \succeq 0$ for any matrix $\mathbf{Z}$. Because $\bar{\sigma}^2 \geq 0$, we have that $\hat{\mathbf{\Pi}} \succeq 0$. ∎

C. Relationship to the Gaussian Process Quadrature MT

The recently proposed Gaussian process quadrature moment transform (GPQMT) [14], together with the BSQMT, are both instances of the general BQ framework. The GPQMT uses a zero-mean GP prior model of the integrand as opposed to the more sophisticated hierarchical prior in eqs. (19) and (21). As a result, the GPQMT weights are affected by the choice of kernel and its parameter values, which is not the case in the BSQMT, where the kernel only affects the last term of the transformed covariance and the weights depend only on the sigma-points and the choice of the function space π. Consequently, this makes BSQMT much less sensitive to misspecification of the kernel parameters, which is a notorious problem plaguing GPQMT. Discussion of the choice of kernel parameters can be found in the original publication [14].

Compared to the zero-mean GP employed in GPQMT, the Bayes–Sard GP is a stronger prior, which means it can provide better fit to the integrand when conditioned on smaller datasets, such as the UT sigma-points, which are especially attractive in nonlinear filtering applications.

D. BSQ Moment Transform in Sigma-Point Filtering

As outlined in Section II, the filtering algorithms use the moment transformations for computing the predictive moments of the system state and measurement. Alg. 2 summarizes the Bayes–Sard quadrature Kalman filter (BSQKF), which employs the proposed BSQ moment transform for computing the predictive moments from Table I. The BSQKF takes two different kernel parameter values, $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_h$, because there are two different functions that need to be integrated (see eqs. (1) and (2)).

Algorithm 2: Bayes-Sard quadrature Kalman filter.

Input: Sequence of measurements $\{\mathbf{z}_k\}_{k=1}^K$, initial conditions $\mathbf{m}_{0|0}^x$, $\mathbf{P}_{0|0}^x$, kernel parameters $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_h$, unit sigma-points $\mathbf{\Xi}$

Output: Sequence of state estimates and covariances $\{\mathbf{m}_{k|k}^x, \mathbf{P}_{k|k}^x\}_{k=1}^K$

for $k \leftarrow 1$ **to** K **do**

```
// predictive state moments
m^x_{k|k-1}, P^x_{k|k-1} ←
BSQMT(f(x_{k-1}), m^x_{k-1|k-1}, P^x_{k-1|k-1}, Ξ, θ_f)
P^x_{k|k-1} ← P^x_{k|k-1} + Q
// predictive measurement moments
m^z_{k|k-1}, P^z_{k|k-1}, P^{xz}_{k|k-1} ←
BSQMT(h(x_k), m^x_{k|k-1}, P^x_{k|k-1}, Ξ, θ_h)
P^z_{k|k-1} ← P^z_{k|k-1} + R
// measurement update (filtering)
// using eqs. (4) and (5)
```

end

VI. Numerical Experiment

In order to test the sigma-point filters based on the BSQ moment transform, we consider the univariate non-stationary growth model (UNGM), which is often used to benchmark particle filters [35]. The system dynamics and the observation model are given by

$$x_k = \frac{1}{2}x_{k-1} + \frac{25x_{k-1}}{1 + x_{k-1}^2} + 8\cos(1.2\,k) + q_{k-1}, \tag{47}$$

$$z_k = \frac{1}{20}x_{k-1}^2 + r_k, \tag{48}$$

with the state noise $q_{k-1} \sim \mathrm{N}(0, 10)$, measurement noise $r_k \sim \mathrm{N}(0, 1)$ and initial conditions $x_0 = x_{0|0} \sim \mathrm{N}(0, 5)$. Kernel scaling used in the BSQ with for the UT ($\kappa = 2$) and the 7-th order GH points was set to $\alpha = 3$ and the lengthscales to $\ell = 0.3$ and $\ell = 0.4$, respectively. For the 5-th order GH points the kernel parameters were set to $\alpha = 5$ and $\ell = 0.6$.

The RMSE was used to measure the tracking performance. The inclination indication (INC) [36], given by

$$\mathrm{INC} = \frac{10}{K}\sum_{k=1}^{K}\log_{10}\frac{(\mathbf{x}_k - \mathbf{m}_{k|k}^x)^\top(\mathbf{P}_{k|k}^x)^{-1}(\mathbf{x}_k - \mathbf{m}_{k|k}^x)}{(\mathbf{x}_k - \mathbf{m}_{k|k}^x)^\top\mathbf{\Sigma}_k^{-1}(\mathbf{x}_k - \mathbf{m}_{k|k}^x)}, \tag{49}$$

where $\mathbf{\Sigma}_k$ is the mean-squared error matrix of the state, was used to measure the credibility of the estimates. A perfectly balanced estimate has $\mathrm{INC} = 0$. For $\mathrm{INC} > 0$, the estimate is said to be optimistic, which is to say the covariance is smaller than it should be, while negative values indicate pessimism. We refer to [36] for other related credibility measures. We simulated the model for $K = 500$ time steps and averaged the performance scores over 100 simulations. The variance of the average scores was estimated by bootstrapping. The parentheses in Tables II and III contain the uncertainty as the least significant digits of 2 standard deviations.

The BSQ filters with classical points were tested against the well-known sigma-point filters as well as the GPQ filters from [14]. As seen in Table II, the filters based on the BSQ outperform the classical sigma-point filters in terms of RMSE. Assuming the GH points are used, BSQKFs can outperform the GPQ filters as well. In comparison with the classical filters, the proposed BSQ filters also provide much more balanced estimates as evidenced by the values of the INC in Table III.

VII. Conclusions and Discussion

In this article, we designed a general-purpose moment transformation based on Bayes–Sard quadrature, which allows for explicit modelling of numerical integration error through the use of a stochastic process model. The hierarchical GP

	Classical	GPQ	BSQ
UT	10.81 (0.14)	10.37 (0.08)	9.70 (0.12)
GH-5	10.03 (0.14)	9.01 (0.11)	8.82 (0.08)
GH-7	9.74 (0.13)	8.80 (0.10)	8.61 (0.09)

Table II: Filter RMSE for the UNGM example.

	Classical	GPQ	BSQ
UT	12.17 (0.06)	4.87 (0.01)	4.57 (0.03)
GH-5	10.33 (0.07)	5.26 (0.03)	1.85 (0.02)
GH-7	9.27 (0.07)	4.95 (0.03)	2.52 (0.03)

Table III: Filter INC for the UNGM example.

prior was shown to be key in developing probabilistic models which lead to the classical quadrature rules used in the sigma-point filters and whose variance is statistically meaningful. We designed the BSQ Kalman filter by leveraging the proposed BSQ moment transform for computation of the predictive moments. Overall, the BSQ-based filters report more balanced estimates and tend to err on the side of caution (the reported estimates are more likely to be pessimistic).

Acknowledgment

JP and OS were supported by the Czech Ministry of Education, Youth and Sports, project LO1506. TK was supported by the Aalto ELEC Doctoral School. TK and CJO were supported by the Lloyd's Register Foundation programme on Data-Centric Engineering at the Alan Turing Institute, UK. SS was supported by the Academy of Finland.

References

[1] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley, 2007.

[2] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*. Wiley-Blackwell, 2012.

[3] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian Multiple Target Tracking*. Artech House, 2014.

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, jun 2006.

[5] S. Gillijns, O. Mendoza, J. Chandrasekar, B. De Moor, D. Bernstein, and A. Ridley, "What is the ensemble Kalman filter and how well does it work?" in *American Control Conference, 2006*, June 2006, p. 6.

[6] R. Bhar, *Stochastic filtering with applications in finance*. World Scientific, 2010.

[7] S. Holmes, G. Klein, and D. W. Murray, "A Square Root Unscented Kalman Filter for visual monoSLAM," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 3710–3716.

[8] K. Lowrey, J. Dao, and E. Todorov, "Real-time State Estimation with Whole-body Multi-contact Dynamics: A Modified UKF Approach," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 1225–1232.

[9] T. Karvonen and S. Särkkä, "Classical quadrature rules via Gaussian processes," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017.

[10] C. J. Oates, S. Niederer, A. Lee, F.-X. Briol, and M. Girolami, "Probabilistic Models for Integration Error in the Assessment of Functional Cardiac Models," in *Advances in Neural Informations Processing Systems (NIPS 2017)*, 2017, pp. 110–118.

[11] F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic, "Probabilistic Integration: A Role in Statistical Computation? (with discussion and rejoinder)," *Statistical Science*, 2018, to appear.

[12] T. Karvonen, S. Särkkä, C. Oates *et al.*, "Symmetry Exploits for Bayesian Cubature Methods," *ArXiv e-prints*, 2018, arXiv:1809.10227.

[13] S. Särkkä, J. Hartikainen, L. Svensson, and F. Sandblom, "On the relation between Gaussian process quadratures and sigma-point methods," *Journal of Advances in Information Fusion*, vol. 11, pp. 31–46, June 2016.

[14] J. Prüher and O. Straka, "Gaussian Process Quadrature Moment Transform," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 2844–2854, Sep. 2018.

[15] T. Karvonen, C. J. Oates, and S. Särkkä, "A Bayes-Sard Cubature Method," *Advances in Neural Informations Processing Systems (NIPS 2018)*, pp. 5882–5893, 2018.

[16] K. Xiong, H. Zhang, and C. W. Chan, "Performance evaluation of UKF-based nonlinear filtering," *Automatica*, vol. 42, no. 2, pp. 261–270, 2006.

[17] K. Xiong, L. D. Liu, and H. Y. Zhang, "Modified unscented Kalman filtering and its application in autonomous satellite navigation," *Aerospace Science and Technology*, vol. 13, no. 4, pp. 238–246, 2009.

[18] T. Karvonen, S. Bonnabel, E. Moulines, and S. Särkkä, "On Stability of a Class of Filters for Non-Linear Stochastic Systems," *ArXiv e-prints*, Sep. 2018, arXiv:1809.05667v1.

[19] A. A. Clifford, *Multivariate error analysis: a handbook of error propagation and calculation in many-parameter systems*. John Wiley & Sons, 1973.

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

[21] I. Arasaratnam, S. Haykin, and R. Elliott, "Discrete-Time Nonlinear Filtering Algorithms Using Gauss-Hermite Quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[22] Y. Wu, D. Hu, M. Wu, and X. Hu, "A Numerical-Integration Perspective on Gaussian Filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2910–2921, 2006.

[23] K. Ito and K. Xiong, "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, 2000.

[24] G. H. Golub and J. H. Welsch, "Calculation of Gauss quadrature rules," *Mathematics of Computation*, vol. 23, no. 106, pp. 221–221, may 1969.

[25] B. Jia, M. Xin, and Y. Cheng, "Sparse-grid quadrature nonlinear filtering," *Automatica*, vol. 48, no. 2, pp. 327–341, 2012.

[26] F. Larkin, "Gaussian measure in hilbert space and applications in numerical analysis," *The Rocky Mountain Journal of Mathematics*, pp. 379–421, 1972.

[27] A. O'Hagan, "Bayes–Hermite quadrature," *Journal of Statistical Planning and Inference*, vol. 29, no. 3, pp. 245–260, 1991.

[28] C. E. Rasmussen and Z. Ghahramani, "Bayesian Monte Carlo," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2003, pp. 505–512.

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

[30] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, ser. Springer Series in Statistics. Springer, 2003.

[31] M. Álvarez, L. Rosasco, and N. Lawrence, "Kernels for Vector-valued Functions: A Review," *Foundations and Trends in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[32] X. Xi, F.-X. Briol, and M. Girolami, "Bayesian Quadrature for Multiple Related Integrals," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 5373–5382.

[33] F. M. Larkin, "Probabilistic error estimates in spline interpolation and quadrature," in *Information Processing 74 (Proceedings of IFIP Congress, Stockholm, 1974)*, vol. 74. North-Holland, 1974, pp. 605–609.

[34] M. Kanagawa, B. K. Sriperumbudur, and K. Fukumizu, "Convergence guarantees for kernel-based quadrature rules in misspecified settings," in *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016, pp. 3288–3296.

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

[36] X. R. Li and Z. Zhao, "Measuring Estimator's Credibility: Noncredibility Index," in *Information Fusion, 2006 9th International Conference on*, 2006, pp. 1–8.