On the Use of Gradient Information in Gaussian Process Quadratures

Jakub Prüher and Simo Särkkä

2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

ON THE USE OF GRADIENT INFORMATION IN GAUSSIAN PROCESS QUADRATURES

Jakub Prüher[*]
University of West Bohemia
Pilsen, Czech Republic
jacobnzw@ntis.zcu.cz

Simo Särkkä[†]
Aalto University
Espoo, Finland
simo.sarkka@aalto.fi

ABSTRACT

Gaussian process quadrature is a promising alternative Bayesian approach to numerical integration, which offers attractive advantages over its well-known classical counterparts. We show how Gaussian process quadrature can naturally incorporate gradient information about the integrand. These results are applied for the design of transformation of means and covariances of Gaussian random variables. We theoretically analyze connections between our proposed moment transform and the linearization transform based on Taylor series. Numerical experiments on common sensor network nonlinearities show that adding gradient information improves the resulting estimates.

Index Terms— Bayesian quadrature, Gaussian process quadrature, moment transformation, derivative, gradient

1. INTRODUCTION

Computation of mean and covariance of a random variable undergoing nonlinear transformation is a problem occurring in trajectory planning [1], multisensor system design [2], uncertainty analysis [3] and other engineering applications. It is also a central problem in many nonlinear Kalman filtering and smoothing algorithms [4, 5, 6]. Due to nonlinearities, many moment transformations resort to numerical approximation of the moment integrals, where the approximation is constructed as a weighted sum of function values at carefully selected design points (sigma-points).

The very prominent classical quadrature rules proceed by approximating the integrated function by a suitable polynomial series and then integrating the approximated function instead. From the polynomial nature of these approximations, it follows, that classical rules integrate polynomials up to a given degree with zero error. In practical applications, however, it is often hard to guarantee that nonlinearities will fall into the polynomial category. Therefore, any quadrature rule inevitably makes approximation errors which, if unacknowledged, can ultimately negatively impact performance of the overall application.

In recent years, Bayesian quadrature (BQ) has become an exciting alternative approach to integration which views numerical quadrature as a problem of statistical inference [7]. The integral is treated as a random variable whose mean estimates the true value while the variance is construed as a model of the numerical integration error. BQ thus produces additional piece information, not present in the classical treatment of quadrature, which can inform the subsequent computations about the achieved integration error.

By Gaussian process quadrature (GPQ) we refer to a special case of BQ in which a Gaussian process (GP) regression is used as a surrogate model of the integrand. Since GP regression is a flexible non-parametric model, it is expected to provide much better approximation capabilities than any fixed-order polynomial series used by the classical rules.

In this paper, we propose the use of function derivatives as a way of decreasing integral variance in GPQ. While the use of derivatives in BQ is not a completely new idea per se, their use has not yet been systematically analyzed in present literature. We design a general GPQ moment transformation which uses gradients as additional source of information about the integrand. We also reveal connections between our proposed transform and the linear transform leveraged by the well-known extended Kalman filter.

2. MOMENT TRANSFORMATIONS

Moment transformation can be formally described as computation of mean and covariance of a random variable

$$\mathbf{y} = \mathbf{g}(\mathbf{x}), \quad \mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{P}), \tag{2.1}$$

where $\mathbf{g} : \mathbb{R}^D \to \mathbb{R}^E$ is a nonlinear function. Typically, probability density of the input variable $\mathbf{x}$ and output variable $\mathbf{y}$ is approximated as jointly Gaussian, which leads to

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{m}\\ \boldsymbol{\mu}\end{bmatrix}, \begin{bmatrix}\mathbf{P} & \mathbf{C}\\ \mathbf{C}^\top & \boldsymbol{\Pi}\end{bmatrix}\right), \tag{2.2}$$

where transformed moments are given by

$$\boldsymbol{\mu} = \mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})], \tag{2.3}$$

$$\boldsymbol{\Pi} = \mathbb{E}_{\mathbf{x}}\left[(\mathbf{g}(\mathbf{x}) - \boldsymbol{\mu})(\mathbf{g}(\mathbf{x}) - \boldsymbol{\mu})^\top\right], \tag{2.4}$$

$$\mathbf{C} = \mathbb{E}_{\mathbf{x}}\left[(\mathbf{x} - \mathbf{m})(\mathbf{g}(\mathbf{x}) - \boldsymbol{\mu})^\top\right], \tag{2.5}$$

where $\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})] = \int \mathbf{g}(\mathbf{x}) \mathcal{N}\left(\mathbf{x} \mid \mathbf{m}, \mathbf{P}\right) \mathrm{d}\mathbf{x}$. In practice, we need to resort to numerical integration schemes which approximate the integrals as weighted sums of function values

$$\mathbb{E}_{\mathbf{x}}[\mathbf{g}(\mathbf{x})] \approx \sum_{n=1}^{N} w_n \mathbf{g}(\mathbf{x}_n), \tag{2.6}$$

where w_n are the quadrature weights and $\mathbf{x}_n$ are the sigma-points (design points). All quadrature-based moment transforms can be

[*]The first author was supported by the Czech Science Foundation, project no. GA 16-19999J and by the project LO1506 of the Czech Ministry of Education, Youth and Sports.

[†]The second author was supported by Academy of Finland.

978-1-5090-0746-2/16/$31.00

written compactly in matrix notation as

$$\boldsymbol{\mu} \approx \mathbf{Y}^\top \mathbf{w}, \tag{2.7}$$

$$\boldsymbol{\Pi} \approx (\mathbf{Y} - \boldsymbol{\mu})^\top \mathbf{W} (\mathbf{Y} - \boldsymbol{\mu}), \tag{2.8}$$

$$\mathbf{C} \approx (\mathbf{X} - \mathbf{m})^\top \mathbf{W}^{\mathrm{c}} (\mathbf{Y} - \boldsymbol{\mu}), \tag{2.9}$$

where $\left[\mathbf{Y}^\top\right]_{*n} = \mathbf{g}(\mathbf{x}_n)$, $\left[(\mathbf{X} - \mathbf{m})^\top\right]_{*n} = \mathbf{x}_n - \mathbf{m}$, and $\left[(\mathbf{Y} - \boldsymbol{\mu})^\top\right]_{*n} = \mathbf{g}(\mathbf{x}_n) - \boldsymbol{\mu}$, where $[\cdot]_{*n}$ denotes the n-th column. The mean weight vector $\mathbf{w} = \begin{bmatrix} w_1 & \dots & w_N \end{bmatrix}^\top$, and $\mathbf{W}$ and $\mathbf{W}^{\mathrm{c}}$ are weight matrices for the covariances. For classical quadratures the weight matrices are diagonal, which is not the case for the GPQ-based transform proposed in Section 5.

A slightly different approach to approximate evaluation of the moment integrals (2.3)–(2.5) is employed by the linearization transform, which relies on Taylor expansion at a single point.

Definition 2.1 (Linearization transform)**.** *The linearization transformation based Gaussian approximation to the joint distribution of* $\mathbf{x}$ *and a transformed random variable* $\mathbf{y} = \mathbf{g}(\mathbf{x})$*, where* $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{P})$*, is given by*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m} \\ \boldsymbol{\mu}_{\mathrm{L}} \end{bmatrix}, \begin{bmatrix} \mathbf{P} & \mathbf{C}_{\mathrm{L}} \\ \mathbf{C}_{\mathrm{L}}^\top & \boldsymbol{\Pi}_{\mathrm{L}} \end{bmatrix} \right) \tag{2.10}$$

where

$$\boldsymbol{\mu}_{\mathrm{L}} = \mathbf{g}(\mathbf{m}), \tag{2.11}$$

$$\boldsymbol{\Pi}_{\mathrm{L}} = \mathbf{G}_{\mathbf{x}}(\mathbf{m}) \mathbf{P} \mathbf{G}_{\mathbf{x}}(\mathbf{m})^\top, \tag{2.12}$$

$$\mathbf{C}_{\mathrm{L}} = \mathbf{P} \mathbf{G}_{\mathbf{x}}(\mathbf{m})^\top \tag{2.13}$$

and $\mathbf{G}_{\mathbf{x}}(\mathbf{m})$ *denotes Jacobian of* $\mathbf{g}$ *evaluated at* $\mathbf{m}$*.*

Evidently, this is an example of a transform which leverages gradient information to calculate the transformed moments. We will refer back to this transform in Section 5, where we reveal connections with our proposed GP quadrature transform.

3. GAUSSIAN PROCESS REGRESSION

This section introduces the problem of GP regression with derivative observations, which is a basis of our proposed moment transform. As a prerequisite, we first outline the basic idea of ordinary GP regression.

The GP regression is a powerful non-parametric regression framework which has been studied extensively [8]. Contrary to parametric models, GP regression offers a flexible way of modeling the unknown function, without relying on rigid parametric structure, and a natural way of expressing the uncertainty about the inferred function.

Suppose we have a dataset consisting of inputs $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_N \end{bmatrix}$ and corresponding noisy function values $\mathbf{y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^\top$ which, for $n = 1, \dots, N$, are given by

$$y_n = g(\mathbf{x}_n) + \varepsilon_n, \qquad \varepsilon_n \sim \mathcal{N}\big(0, \sigma_\varepsilon^2\big), \tag{3.1}$$

where $g : \mathbb{R}^D \to \mathbb{R}$ is the unknown functional relationship we wish to infer. GP regression framework assumes that this function is a priori distributed according to a Gaussian process

$$g(\mathbf{x}) \sim \mathcal{GP}(0,\, k(\mathbf{x}, \mathbf{x}')), \tag{3.2}$$

where $k(\mathbf{x}, \mathbf{x}')$ is the covariance function (kernel). From the Bayesian perspective, by choosing a kernel we are putting a prior on the functions themselves and by doing so introduce assumptions about the function's behavior. The zero mean is chosen for analytical convenience without loss of generality. Let $\mathcal{D} = \{(\mathbf{x}_n,\, y_n)\}_{n=1}^N$ denote available data, then the posterior GP mean and predictive GP variance are given by

$$\mathbb{E}_g[g(\mathbf{x}) \mid \mathcal{D}] = \mathbf{k}^\top(\mathbf{x})(\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I})^{-1} \mathbf{y}, \tag{3.3}$$

$$\mathbb{V}_g[g(\mathbf{x}) \mid \mathcal{D}] = k(\mathbf{x},\, \mathbf{x}) - \mathbf{k}^\top(\mathbf{x})(\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{x}), \tag{3.4}$$

where $\left[\mathbf{k}^\top(\mathbf{x})\right]_{*n} = k(\mathbf{x},\, \mathbf{x}_n)$ and $[\mathbf{K}]_{mn} = k(\mathbf{x}_m,\, \mathbf{x}_n)$ is the kernel matrix, where $[\cdot]_{mn}$ denotes the matrix element at position (m, n). The posterior GP mean (3.3) serves effectively as an approximation of $g(\mathbf{x})$ while predictive variance (3.4) informs about the function value uncertainty.

3.1. Derivative Observations in GP Regression

The use of derivative observations in GP regression is a special case of using linear operator observations which has been previously discussed, for example, in [9, 10]. We briefly review the problem setup and state the main results.

Consider the GP regression problem as described above, where, in addition to observing function values y_n for every input $\mathbf{x}_n$, we also observe gradients

$$\boldsymbol{\delta}_n = \mathbf{d}(\mathbf{x}_n) + \boldsymbol{\varepsilon}_n, \qquad \boldsymbol{\varepsilon}_n \sim \mathcal{N}\big(\mathbf{0},\, \sigma_\varepsilon^2 \mathbf{I}\big), \tag{3.5}$$

where $\mathbf{d} : \mathbb{R}^D \to \mathbb{R}^D$ is a gradient of function $g(\mathbf{x})$ defined as

$$\mathbf{d}(\mathbf{x}) \coloneqq \begin{bmatrix} \frac{\partial g}{\partial x_1} & \cdots & \frac{\partial g}{\partial x_D} \end{bmatrix}^\top. \tag{3.6}$$

Since gradient is a linear operator acting on a GP distributed function g, the $\mathbf{d}$ is also a GP. This fact is an infinite dimensional analogue of the affine property of Gaussian random variables. With gradient observations incorporated, our dataset is now $\tilde{\mathcal{D}} = \{(\mathbf{x}_n,\, y_n,\, \boldsymbol{\delta}_n)\}_{n=1}^N$ and the posterior mean and predictive variance are given by

$$\mathbb{E}_g\left[g(\mathbf{x}) \mid \tilde{\mathcal{D}}\right] = \widetilde{\mathbf{k}}^\top(\mathbf{x})(\widetilde{\mathbf{K}} + \sigma_\varepsilon^2 \mathbf{I})^{-1} \widetilde{\mathbf{y}}, \tag{3.7}$$

$$\mathbb{V}_g\left[g(\mathbf{x}) \mid \tilde{\mathcal{D}}\right] = k(\mathbf{x},\, \mathbf{x}) - \widetilde{\mathbf{k}}^\top(\mathbf{x})(\widetilde{\mathbf{K}} + \sigma_\varepsilon^2 \mathbf{I})^{-1} \widetilde{\mathbf{k}}(\mathbf{x}), \tag{3.8}$$

where all observations are arranged in a $(N + ND \times 1)$ vector $\widetilde{\mathbf{y}} = \mathrm{vec}\left(\begin{bmatrix} \mathbf{y} & \boldsymbol{\delta}_1 & \dots & \boldsymbol{\delta}_N \end{bmatrix}\right)$ and vec is a vectorization operator. The layout of the block matrices is

$$\widetilde{\mathbf{k}}(\mathbf{x}) = \begin{bmatrix} \mathbf{k}(\mathbf{x}) \\ \mathbf{k}_{g\mathbf{d}}(\mathbf{x}) \end{bmatrix}, \quad \text{and} \quad \widetilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K} & \mathbf{K}_{g\mathbf{d}} \\ \mathbf{K}_{\mathbf{d}g} & \mathbf{K}_{\mathbf{dd}} \end{bmatrix}, \tag{3.9}$$

with $\mathbf{K}_{\mathbf{d}g} = \mathbf{K}_{g\mathbf{d}}^\top$ and matrices $\mathbf{k}^\top(\mathbf{x})$ and $\mathbf{K}$ defined above. Let $\otimes$ denote Kronecker product, then $\mathbf{k}_{g\mathbf{d}}(\mathbf{x}) = \sum_{n=1}^N \mathbf{e}_n \otimes k_{g\mathbf{d}}(\mathbf{x},\, \mathbf{x}_n)$, $\mathbf{K}_{g\mathbf{d}} = \sum_{m,n=1}^N \mathbf{e}_m \mathbf{e}_n^\top \otimes k_{g\mathbf{d}}(\mathbf{x}_m,\, \mathbf{x}_n)$ and $\mathbf{K}_{\mathbf{dd}} = \sum_{m,n=1}^N \mathbf{e}_m \mathbf{e}_n^\top \otimes k_{\mathbf{dd}}(\mathbf{x}_m,\, \mathbf{x}_n)$. Since

$$k_{g\mathbf{d}}(\mathbf{x},\, \mathbf{x}_n) \coloneqq \frac{\partial}{\partial \mathbf{x}'}\, k\big(\mathbf{x},\, \mathbf{x}'\big)\bigg|_{\mathbf{x}'=\mathbf{x}_n}, \tag{3.10}$$

$$k_{\mathbf{dd}}(\mathbf{x}_m,\, \mathbf{x}_n) \coloneqq \frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}'} k\big(\mathbf{x},\, \mathbf{x}'\big)\bigg|_{\mathbf{x}=\mathbf{x}_m, \mathbf{x}'=\mathbf{x}_n}, \tag{3.11}$$

are a $(1 \times D)$ vector and $(D \times D)$ matrix respectively, the $\mathbf{K}_{g\mathbf{d}}$ is $(N \times DN)$ block matrix and $\mathbf{K}_{\mathbf{dd}}$ is $(DN \times DN)$ block matrix. The Figure 1b shows the reduced predictive variance of GP regression fit when gradient observations are used. The same kernel hyperparameters are used in both cases.

It is easy to show that the predictive variance is always decreased by including additional information from gradients. The proof is omitted for space reasons.

4. GAUSSIAN PROCESS QUADRATURE

Bayesian quadrature [7, 11] offers a different perspective on the problem of quadrature. The integral is seen as a random variable, with mean representing the numerically approximated value and variance modeling the integration error. In the following, for brevity reasons, we drop the conditioning on $\mathcal{D}$ in $\mathbb{E}_{g|\mathcal{D}}[\,\cdot\,]$. Since function can be evaluated exactly, the observation noise is formally zero. The posterior integral mean and integral variance are [12]

$$\mathbb{E}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbb{E}_{\mathbf{x}}[\mathbb{E}_g[g(\mathbf{x})]], \tag{4.1}$$

$$\mathbb{V}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbb{E}_{\mathbf{x},\mathbf{x}'}\left[\mathbb{C}_g\left[g(\mathbf{x}),\, g(\mathbf{x}')\right]\right], \tag{4.2}$$

where the posterior GP covariance is given by

$$\mathbb{C}_g\left[g(\mathbf{x}),\, g(\mathbf{x}')\right] = k\big(\mathbf{x},\, \mathbf{x}'\big) - \mathbf{k}^\top(\mathbf{x})\mathbf{K}^{-1}\mathbf{k}\big(\mathbf{x}'\big) \tag{4.3}$$

with $\mathbb{C}$ denoting the covariance operator. Plugging (3.3) and (4.3) into (4.1) and (4.2), we get

$$\mathbb{E}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbb{E}_{\mathbf{x}}\left[\mathbf{k}^\top(\mathbf{x})\right]\mathbf{K}^{-1}\mathbf{y} = \mathbf{q}^\top\mathbf{K}^{-1}\mathbf{y}, \tag{4.4}$$

$$\mathbb{V}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \overline{\overline{k}} - \mathbf{q}^\top\mathbf{K}^{-1}\mathbf{q}, \tag{4.5}$$

where $\overline{\overline{k}} = \mathbb{E}_{\mathbf{x},\mathbf{x}'}[k(\mathbf{x},\, \mathbf{x}')]$. Notice we can define quadrature weights as $\mathbf{w}^\top = \mathbf{q}^\top\mathbf{K}^{-1}$ which shows that (4.4) is indeed a quadrature rule.

4.1. Gradient Information in GP Quadrature

In this section, we propose the use of gradient information in GP quadrature by using the expressions for GP regression with gradient observations from Subsection 3.1. Gradient observations are incorporated by plugging (3.7) and (3.8) into (4.1) and (4.2), which yields

$$\mathbb{E}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbb{E}_{\mathbf{x}}\left[\widetilde{\mathbf{k}}^\top(\mathbf{x})\right]\widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{y}} = \widetilde{\mathbf{q}}^\top\widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{y}}, \tag{4.6}$$

$$\mathbb{V}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \overline{\overline{k}} - \widetilde{\mathbf{q}}^\top\widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{q}}, \tag{4.7}$$

Since $\widetilde{\mathbf{y}}$ contains gradient observations, note that (4.6) is now a sum of a linear combination of function values and gradients. The Figure 1c compares the resulting Gaussian distributions over the value of the integral. Both the GP quadrature and GP quadrature with gradient observations produce distributions centered on the true value of the integral. Conditioning on additional gradient observations decreases integral variance.

5. GPQ TRANSFORMS WITH GRADIENTS

General GP quadrature moment transform relies on the following approximation to the true moments

$$\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})] \approx \mathbb{E}_{g,\mathbf{x}}[g(\mathbf{x})], \tag{5.1}$$

$$\mathbb{V}_{\mathbf{x}}[g(\mathbf{x})] \approx \mathbb{V}_{g,\mathbf{x}}[g(\mathbf{x})], \tag{5.2}$$

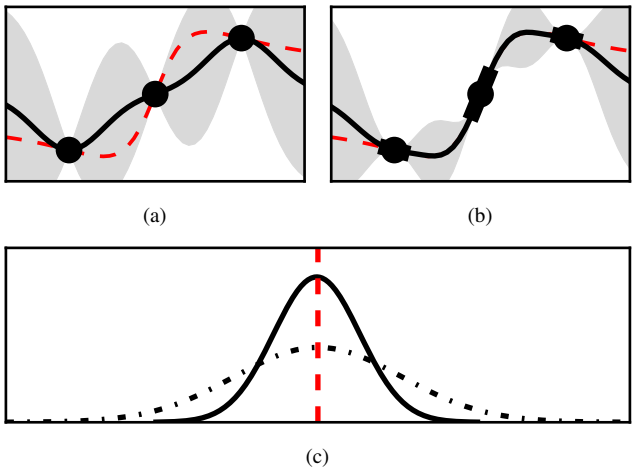


Fig. 1. Approximation of the true function (dashed) with the GP mean function (solid). (a) Approximation using function observations (dots) only. (b) Approximation using function values and gradient observations (line segments). (c) Densities over the value of the integral. The integral variance of GPQ+D (solid) is visibly smaller than that of GPQ (dash dot). Both densities concentrate near the true value of the integral (dashed).

where the approximate expectations account for the uncertainty in the function g. We recognize that $\mathbb{E}_{g,\mathbf{x}}[g(\mathbf{x})] = \mathbb{E}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbb{E}_{\mathbf{x}}[\mathbb{E}_g[g(\mathbf{x})]]$, which means that the mean of integral is equivalent to integral of the mean function. Furthermore, the expression for transformed variance can be expanded as

$$\mathbb{V}_{g,\mathbf{x}}[g(\mathbf{x})] = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_g[g(\mathbf{x})]^2\right] - \mathbb{E}_{\mathbf{x}}[\mathbb{E}_g[g(\mathbf{x})]]^2 + \mathbb{E}_{\mathbf{x}}[\mathbb{V}_g[g(\mathbf{x})]], \tag{5.3}$$

which is the same expression used for making GP predictions at uncertain inputs [13, 14] and can be shown [15] to contain the variance of the mean integral $\mathbb{V}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]]$.

The stochastic decoupling substitution is often used in recursive filtering and smoothing applications to express the moment integrals (2.3)–(2.5) in terms of decoupled random variable $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0},\, \mathbf{I})$. Note that, so far, we have formulated GPQ for scalar functions only. Integration of a vector functions (2.1) where $\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g^1(\mathbf{x}) & \dots & g^E(\mathbf{x}) \end{bmatrix}$, can be handled by applying the procedure to every output dimension independently. The following definition summarizes the proposed GPQ-based moment transform with gradient observations.

Definition 5.1 (General GPQ+D moment transform)**.** *The general Gaussian process quadrature based Gaussian approximation to the joint distribution of* $\mathbf{x}$ *and a transformed random variable* $\mathbf{y} = \mathbf{g}(\mathbf{x})$*, where* $\mathbf{x} \sim \mathcal{N}(\mathbf{m},\, \mathbf{P})$*, is given by*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m} \\ \boldsymbol{\mu}_{\mathrm{G}} \end{bmatrix},\, \begin{bmatrix} \mathbf{P} & \mathbf{C}_{\mathrm{G}} \\ \mathbf{C}_{\mathrm{G}}^\top & \boldsymbol{\Pi}_{\mathrm{G}} \end{bmatrix}\right) \tag{5.4}$$

$$\boldsymbol{\mu}_{\mathrm{G}} = \widetilde{\mathbf{Y}}^\top\mathbf{w} \tag{5.5}$$

$$\boldsymbol{\Pi}_{\mathrm{G}} = \widetilde{\mathbf{Y}}^\top\mathbf{W}\widetilde{\mathbf{Y}} - \boldsymbol{\mu}_{\mathrm{G}}\boldsymbol{\mu}_{\mathrm{G}}^\top + \bar{\sigma}_g\mathbf{I} \tag{5.6}$$

$$\mathbf{C}_{\mathrm{G}} = \mathbf{W}^{\mathrm{c}}\widetilde{\mathbf{Y}} - \mathbf{m}\boldsymbol{\mu}_{\mathrm{G}}^\top \tag{5.7}$$

$$\bar{\sigma}_g = \bar{k} - \mathrm{tr}\left(\widetilde{\mathbf{Q}}\widetilde{\mathbf{K}}^{-1}\right) \tag{5.8}$$

where $\left[\widetilde{\mathbf{Y}}\right]_{*e} = \operatorname{vec}\left(\begin{bmatrix}\mathbf{y}^e & \boldsymbol{\delta}_1^e & \dots & \boldsymbol{\delta}_N^e\end{bmatrix}\right)$ *are function values and gradient observations of the* e*-th output dimension of* $\mathbf{g}(\mathbf{x})$*. The augmented kernel matrix* $\left[\widetilde{\mathbf{K}}\right]_{mn}$ *is defined in* (3.9) *and the weight matrices are* $\mathbf{w}^\top = \widetilde{\mathbf{q}}^\top \widetilde{\mathbf{K}}^{-1}$, $\mathbf{W} = \widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{Q}}\widetilde{\mathbf{K}}^{-1}$, $\mathbf{W}^{\mathrm{c}} = \widetilde{\mathbf{R}}\widetilde{\mathbf{K}}^{-1}$*. The block matrices are*

$$\widetilde{\mathbf{q}} = \begin{bmatrix}\mathbf{q} \\ \mathbf{q}_{\mathbf{d}}\end{bmatrix}, \quad \widetilde{\mathbf{Q}} = \begin{bmatrix}\mathbf{Q} & \mathbf{Q}_{g\mathbf{d}} \\ \mathbf{Q}_{\mathbf{d}g} & \mathbf{Q}_{\mathbf{dd}}\end{bmatrix}, \quad \widetilde{\mathbf{R}} = \begin{bmatrix}\mathbf{R} & \mathbf{R}_{\mathbf{d}}\end{bmatrix} \tag{5.9}$$

where

$$\mathbf{q} = \sum \mathbf{e}_n \mathbb{E}_{\mathbf{x}}[k(\mathbf{x}, \mathbf{x}_n)], \qquad \mathbf{q}_{\mathbf{d}} = \sum \mathbf{e}_n \otimes \mathbb{E}_{\mathbf{x}}[k_{g\mathbf{d}}(\mathbf{x}, \mathbf{x}_n)], \tag{5.10}$$

$$\mathbf{R} = \sum \mathbf{e}_n \mathbb{E}_{\mathbf{x}}[\mathbf{x}k(\mathbf{x}, \mathbf{x}_n)], \quad \mathbf{R}_{\mathbf{d}} = \sum \mathbf{e}_n \otimes \mathbb{E}_{\mathbf{x}}[\mathbf{x}k_{g\mathbf{d}}(\mathbf{x}, \mathbf{x}_n)]. \tag{5.11}$$

and

$$\mathbf{Q} = \sum \mathbf{e}_m \mathbf{e}_n^\top \mathbb{E}_{\mathbf{x}}[k(\mathbf{x}_m, \mathbf{x})k(\mathbf{x}, \mathbf{x}_n)], \tag{5.12}$$

$$\mathbf{Q}_{g\mathbf{d}} = \sum \mathbf{e}_m \mathbf{e}_n^\top \otimes \mathbb{E}_{\mathbf{x}}[k(\mathbf{x}_m, \mathbf{x})k_{g\mathbf{d}}(\mathbf{x}, \mathbf{x}_n)], \tag{5.13}$$

$$\mathbf{Q}_{\mathbf{dd}} = \sum \mathbf{e}_m \mathbf{e}_n^\top \otimes \mathbb{E}_{\mathbf{x}}[k_{\mathbf{d}g}(\mathbf{x}_m, \mathbf{x})k_{g\mathbf{d}}(\mathbf{x}, \mathbf{x}_n)]. \tag{5.14}$$

where the summations are over $m, n = 1, \dots, N$ *and* $\mathbf{x}_n$ *are* arbitrarily *chosen sigma-points.*

5.1. Connections of GPQ+D to linearization

General GPQ+D moment transform is formulated for *arbitrary* kernel and sigma-point sets. Below we give proofs that GPQ+D reduces to linearized transform for two particular choices of the kernel. In the derivations, we use the centered variant of the GPQ+D transform, which uses a substitution $\mathbf{x} = \mathbf{m} + \boldsymbol{\eta}$, $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$ in the Gaussian integrals (2.3)–(2.5). Thus GP models the function $\tilde{g}(\boldsymbol{\eta}) = g(\mathbf{x}) = g(\mathbf{m} + \boldsymbol{\eta})$, so that $\tilde{g}(\boldsymbol{\eta}) \sim \mathcal{GP}(0, k(\boldsymbol{\eta}, \boldsymbol{\eta}'))$. Expressions for the mean (5.5) and covariance (5.6) of the centered GPQ+D remain formally the same except for the input-output covariance which becomes

$$\mathbf{C}_{\mathrm{G}} = \mathbf{W}^{\mathrm{c}}\mathbf{Y}. \tag{5.15}$$

In order for the GPQ+D to be equivalent to the linearization transform, clearly, (5.5), (5.6) and (5.15) should be equal to (2.11), (2.12) and (2.13) respectively. This can be achieved when GPQ+D uses single sigma-point $\mathbf{x}_0 = \mathbf{m}$ ($\boldsymbol{\eta}_0 = \mathbf{0}$), in which case $\mathbf{Y} = \begin{bmatrix}\mathbf{g}(\mathbf{m}) & \mathbf{G}_{\mathbf{x}}(\mathbf{m})\end{bmatrix}^\top$ and the weights are given by

$$\mathbf{w} = \begin{bmatrix}1 \\ \mathbf{0}\end{bmatrix}, \ \mathbf{W} = \begin{bmatrix}1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{P}\end{bmatrix}, \ \mathbf{W}^{\mathrm{c}} = \begin{bmatrix}0 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{P}\end{bmatrix}. \tag{5.16}$$

Additionally, $\bar{\sigma}_g = 0$ must hold.

Linearized transform uses derivative at one point to approximate the nonlinearity with a tangent line. The same can be achieved with GP regression using affine kernel and gradient observations.

Theorem 5.1 (GPQ+D with affine kernel)**.** *Let* $k(\boldsymbol{\eta}, \boldsymbol{\eta}') = \sigma_0^2 + \boldsymbol{\eta}^\top \boldsymbol{\Sigma} \boldsymbol{\eta}'$ *with* $\boldsymbol{\Sigma} = \operatorname{diag}\left(\begin{bmatrix}\sigma_1^2 & \dots & \sigma_D^2\end{bmatrix}\right)$*. Assume that one sigma-point* $\boldsymbol{\eta}_0 = \mathbf{0}$ *is used. Then the general GPQ+D transform reduces to the linearized transform and the variance of the mean integral is zero.*

Proof. The kernel derivatives are

$$k_{g\mathbf{d}}(\boldsymbol{\eta}, \boldsymbol{\eta}') = \boldsymbol{\eta}^\top \boldsymbol{\Sigma}, \tag{5.17}$$

$$k_{\mathbf{dd}}(\boldsymbol{\eta}, \boldsymbol{\eta}') = \boldsymbol{\Sigma}. \tag{5.18}$$

After evaluating the kernel at $\boldsymbol{\eta}_0$ and its expectations, we find that

$$\widetilde{\mathbf{q}} = \begin{bmatrix}\sigma_0^2 \\ \mathbf{0}\end{bmatrix}, \ \widetilde{\mathbf{K}}^{-1} = \begin{bmatrix}\sigma_0^{-2} & \mathbf{0}^\top \\ \mathbf{0} & \boldsymbol{\Sigma}^{-1}\end{bmatrix}, \ \widetilde{\mathbf{Q}} = \begin{bmatrix}\sigma_0^4 & \mathbf{0}^\top \\ \mathbf{0} & \boldsymbol{\Sigma}\mathbf{P}\boldsymbol{\Sigma}\end{bmatrix} \tag{5.19}$$

plugging into the expressions for weight matrices $\mathbf{w}$, $\mathbf{W}$ and $\mathbf{W}^{\mathrm{c}}$ from Definition 5.1, we obtain the linear transform weights given by (5.16). Expected GP variance reduces to $\bar{\sigma}_g = \sigma_0^2 + \operatorname{tr}(\mathbf{P}\boldsymbol{\Sigma}) - (\sigma_0^2 + \operatorname{tr}(\mathbf{P}\boldsymbol{\Sigma})) = 0$. Plugging $\widetilde{\mathbf{q}}$ and $\widetilde{\mathbf{K}}^{-1}$ into (4.7), it is easily verified that the variance of the mean integral is zero. □

In a similar fashion, we can use the radial basis function (RBF) kernel with infinite lengthscale to obtain linearized transform.

Theorem 5.2 (GPQ+D with RBF kernel and $\ell \to \infty$)**.** *Let*

$$k(\boldsymbol{\eta}, \boldsymbol{\eta}') = \alpha^2 \exp\left(-\tfrac{1}{2}(\boldsymbol{\eta} - \boldsymbol{\eta}')^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{\eta} - \boldsymbol{\eta}')\right)$$

where $\boldsymbol{\Lambda} = \ell^2\mathbf{I}$ *and* ℓ *is the lengthscale hyper-parameter. Assume that one sigma-point* $\boldsymbol{\eta}_0 = \mathbf{0}$ *is used. Then the GPQ+D transform reduces to the linear transform for* $\ell \to \infty$ *and variance of the mean integral is* $\alpha^2 - 1$*.*

Proof. The kernel derivatives are

$$k_{g\mathbf{d}}(\boldsymbol{\eta}, \boldsymbol{\eta}') = \boldsymbol{\Lambda}^{-1}(\boldsymbol{\eta} - \boldsymbol{\eta}')k(\boldsymbol{\eta}, \boldsymbol{\eta}') \tag{5.20}$$

$$k_{\mathbf{dd}}(\boldsymbol{\eta}, \boldsymbol{\eta}') = \left[\mathbf{I} - \boldsymbol{\Lambda}^{-1}(\boldsymbol{\eta} - \boldsymbol{\eta}')(\boldsymbol{\eta} - \boldsymbol{\eta}')^\top\right]\boldsymbol{\Lambda}^{-1}k(\boldsymbol{\eta}, \boldsymbol{\eta}') \tag{5.21}$$

After computing all the necessary kernel expectations (5.10)–(5.14), we find that

$$\widetilde{\mathbf{q}}^\top = \begin{bmatrix}\alpha^2\left|\mathbf{P}\boldsymbol{\Lambda}^{-1} + \mathbf{I}\right|^{-1/2} & \mathbf{0}\end{bmatrix} \tag{5.22}$$

$$\widetilde{\mathbf{K}}^{-1} = \begin{bmatrix}\alpha^{-2} & \mathbf{0}^\top \\ \mathbf{0} & \alpha^{-2}\boldsymbol{\Lambda}\end{bmatrix} \tag{5.23}$$

$$\widetilde{\mathbf{Q}} = \begin{bmatrix}b & \mathbf{0}^\top \\ \mathbf{0} & b\boldsymbol{\Lambda}^{-1}(2\boldsymbol{\Lambda}^{-1} + \mathbf{P}^{-1})^{-1}\boldsymbol{\Lambda}^{-1}\end{bmatrix} \tag{5.24}$$

and $b = \alpha^4\left|2\boldsymbol{\Lambda}^{-1}\mathbf{P} + \mathbf{I}\right|^{-1/2}$. The weights are

$$\mathbf{w} = \left|\boldsymbol{\Lambda}^{-1}\mathbf{P} + \mathbf{I}\right|^{-1/2}\begin{bmatrix}1 & \mathbf{0}^\top\end{bmatrix}^\top, \tag{5.25}$$

$$\mathbf{W} = \left|2\boldsymbol{\Lambda}^{-1}\mathbf{P} + \mathbf{I}\right|^{-1/2}\begin{bmatrix}1 & \mathbf{0}^\top \\ \mathbf{0} & (2\boldsymbol{\Lambda}^{-1} + \mathbf{P}^{-1})^{-1}\end{bmatrix}, \tag{5.26}$$

$$\mathbf{W}^{\mathrm{c}} = \left|\boldsymbol{\Lambda}^{-1}\mathbf{P} + \mathbf{I}\right|^{-1/2}\begin{bmatrix}0 & \mathbf{0}^\top \\ \mathbf{0} & (\boldsymbol{\Lambda}^{-1} + \mathbf{P}^{-1})^{-1}\end{bmatrix}. \tag{5.27}$$

Taking a limit for $\ell \to \infty$, we obtain the linear transform weights (5.16). Expected GP variance is $\bar{\sigma}_g = \alpha^2 - b\left[\alpha^{-2} + \operatorname{tr}\left((2\boldsymbol{\Lambda}^{-1} + \mathbf{P}^{-1})^{-1}\boldsymbol{\Lambda}^{-1}\right)\right]$ for which $\lim_{\ell\to\infty}\bar{\sigma}_g = 0$. Plugging in $\widetilde{\mathbf{q}}$ and $\widetilde{\mathbf{K}}^{-1}$ into (4.7), the variance of the mean integral becomes

$$\mathbb{V}_g[\mathbb{E}_{\boldsymbol{\eta}}[g(\boldsymbol{\eta})]] = \alpha^2\left|2\boldsymbol{\Lambda}^{-1}\mathbf{P} + \mathbf{I}\right|^{-1/2} - \left|\boldsymbol{\Lambda}^{-1}\mathbf{P} + \mathbf{I}\right|^{-1},$$

which, for $\ell \to \infty$, approaches $\alpha^2 - 1$. □

6. NUMERICAL EXPERIMENTS

In this section we test our proposed methods experimentally. In all numerical experiments we tested decoupled variants of GPQ and GPQ+D moment transforms. Both transforms operate with the RBF kernel and use spherical-radial unit sigma-points. The kernel hyper-parameters ℓ and α were set to good heuristic values for each experiment independently. Both transforms were compared against the spherical-radial transform (SR) [16].

6.1. Analytical example

In the first experiment we considered a simple example where the transformed moments can be computed exactly. A well known fact is that a random variable given by sum of squares function

$$z = g_{\text{SOS}}(\mathbf{x}) = \mathbf{x}^\top \mathbf{x}, \quad \text{where} \quad \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D) \tag{6.1}$$

has density $z \sim \chi^2(D)$ with mean D and variance $2D$. Kernel hyper-parameters of both GPQ-based transforms were $\alpha = 1$ and $\ell = 10$. It is apparent from the Table 1 that, while SR and GPQ transforms

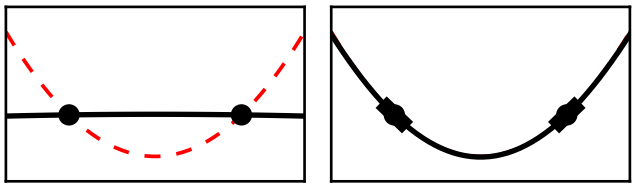
D	1	5	10	25
Mean				
True	1.00	5.00	10.00	25.00
SR	1.00	4.95	10.00	25.00
GPQ	1.00	5.00	10.00	25.02
GPQ+D	0.99	5.00	9.89	24.49
Variance				
True	2.00	10.00	20.00	50.00
SR	0.00	0.00	0.00	0.00
GPQ	0.00	0.01	0.05	0.78
GPQ+D	**1.92**	**9.61**	**19.16**	**46.44**

Table 1. Comparison of transformed mean and variance for increasing dimension D computed by the SR, GPQ and GPQ+D moment transforms.

capture the mean fairly accurately, they completely fail to capture the variance of transformed random variable. GPQ+D is the only transform that comes close to the true transformed variances. Table 2 demonstrates that by including gradient information the variance of the mean integral decreases. Figure 2 shows the GP regression fit to (6.1) for $D = 1$. The ordinary GP regression fit, in Figure 2a, fails to capture the true function, whereas by including the derivative information the fit improves significantly as seen in Figure 2b. This is because we are using symmetric sigma-point set and (6.1) is an even function.

D	1	5	10	25
GPQ	1.14e-08	5.56e-08	3.31e-07	4.62e-06
GPQ+D	**6.62e-09**	**3.16e-08**	**2.75e-07**	**4.27e-06**

Table 2. Comparison of variance of the mean integral for GPQ and GPQ+D. Overall, including derivative information decreases variance



(a) (b)

Fig. 2. (a) Approximation used by GPQ. (b) Approximation used by GPQ+D.

6.2. Sensor network measurements

The second experiment is inspired by [17], where authors considered nonlinear measurements which are commonly encountered in sensor networks. These are, time of arrival (TOA), direction of arrival (DOA) and received signal strength (RSS) given by

$$g_{\text{TOA}}(\mathbf{x}) = \|\mathbf{x}\|_2, \tag{6.2}$$

$$g_{\text{DOA}}(\mathbf{x}) = \text{atan2}(x_1, x_2), \tag{6.3}$$

$$g_{\text{RSS}}(\mathbf{x}) = 10 - 20\log_{10}(\|\mathbf{x}\|_2^2), \tag{6.4}$$

where atan2 is the four-quadrant variant of atan. As an example of vector function, we considered radar measurements (RDR) which arise as a mapping of range r and bearing θ to Cartesian coordinates given by

$$g_{\text{RDR}}(\mathbf{x}) = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix}. \tag{6.5}$$

The symmetrized KL-divergence was used to measure distance between the baseline distribution, computed by Monte Carlo transform with 20,000 samples, and the transformed distribution computed by SR, GPQ and GPQ+D respectively. The covariance of the input distribution was randomly generated and results were averaged over 100 MC simulations. Since DOA and RDR functions are limited to two-dimensional inputs, we tested for $D = 2$ only. Table 4 shows that including gradient observations in GPQ can improve the average symmetrized KL-divergence; in case of TOA, even by two orders of magnitude. Values of the kernel hyper-parameters for individual

	TOA	RSS	DOA	RDR
ℓ	3.0	0.2	2.0	5.0
α	10	10	1	1

Table 3. Values of the RBF kernel hyper-parameters.

functions are summarized in Table 3.

	SR	GPQ	GPQ+D
TOA	2.74e-02	3.37e-01	**4.61e-03**
RSS	4.48e+00	4.76e-01	**4.70e-01**
DOA	5.48e-03	5.99e-03	**1.80e-03**
RDR	6.48e-01	7.07e-01	**2.94e-01**

Table 4. Comparison of the SR, GPQ and GPQ+D moment transforms in terms of symmetrized KL-divergence performance.

7. CONCLUSION

In this article, we have analyzed the use of gradient observations in GP quadratures and designed general moment transformations based on GP quadrature with gradients. The proposed transforms were tested on a range of functions arising in the sensor network applications. Using Monte Carlo simulations we have shown that including additional gradient observations improves the results. Finally, we gave proofs for two limit cases when the proposed transform reduces to the linear transform based on Taylor series.

8. REFERENCES

[1] Hao Sun, Weiwen Deng, Sumin Zhang, Shanshan Wang, and Yutan Zhang, "Trajectory planning for vehicle autonomous driving with uncertainties," in *Informative and Cybernetics for Computational Social Systems (ICCSS), 2014 International Conference on*, Oct 2014, pp. 34–38.

[2] H. Zangl and G. Steiner, "Optimal design of multiparameter multisensor systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1484–1491, July 2008.

[3] A. A. Savin, V. G. Guba, and B. D. Maxson, "Covariance based uncertainty analysis with unscented transformation," in *Microwave Measurement Conference, 2013 82nd ARFTG*, Nov 2013, pp. 1–4.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, Wiley-Blackwell, 2001.

[5] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge University Press, New York, 2013.

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

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

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

[9] R. Murray-Smith and B. A. Pearlmutter, "Transformations of Gaussian process priors," in *Deterministic and Statistical Methods in Machine Learning*, pp. 110–123. Springer, 2005.

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

[11] P. Diaconis, "Bayesian numerical analysis," in *Statistical Decision Theory and Related Topics IV*, S. S. Gupta and J. O. Berger, Eds. 1988, pp. 163–175, Springer.

[12] 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. 2003, number 1, pp. 505–512, MIT Press.

[13] A. Girard, C. E. Rasmussen, J. Quiñonero Candela, and R. Murray-Smith, "Gaussian Process Priors With Uncertain Inputs Application to Multiple-Step Ahead Time Series Forecasting," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. 2003, pp. 545–552, MIT Press.

[14] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic moment-based Gaussian process filtering," in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. 2009, pp. 1–8, ACM Press.

[15] J. Prüher and M. Šimandl, "Bayesian Quadrature in Nonlinear Filtering," in *Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on*, July 2015, vol. 01, pp. 380–387.

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

[17] 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.