Nonlinear Filtering Using the Double Exponential Transformation

Quade Butler[a], Brett Sicard[a], Waleed Hilal[a], S. Andrew Gadsden[a], and Youssef Ziada[b]

[a]McMaster University, 1280 Main St West, Hamilton, Canada
[b]Ford Motor Company, 36200 Plymouth Road, Livonia, USA

ABSTRACT

Nonlinear Gaussian filters have traditionally used cubature rules and/or Gaussian quadrature to compute multidimensional expectation integrals recursively, which provide mean and covariance estimates of the state and state error, respectively. Minimal and near minimal-point filters attain moderate accuracy while avoiding the so-called "curse of dimensionality", but their accuracy can diverge over time. Recent trends in cubature-based filtering have opted for more evaluation points to increase accuracy at the cost of higher computational overhead, while still avoiding the dreaded curse. These methods use more complex and higher-degree cubature rules. The present work, contrary to recent trends, uses a quadrature method other than that of the Gaussian variety. Double exponential quadrature is used to achieve high levels of relative accuracy with a moderate number of evaluation points, rivalling that of current state-of-the-art Gaussian filters and the best-in-class Gauss-Hermite filter.

Keywords: Nonlinear filtering, Gaussian filtering, Kalman filtering, Cubature Kalman filter, Double exponential quadrature

1. INTRODUCTION

Gaussian filtering is the estimation of hidden states and/or parameters of a nonlinear dynamic system subjected to Gaussian noise processes from uncertain measurements. Contributions to the field of estimation date back to the fifteenth century, but much of the modern literature on Gaussian filtering, in particular filtering from a Bayesian perspective, occurred within the last 30 years.[1] The Bayesian paradigm of filtering includes calculating the conditional *a posteriori* probability density function (PDF) of the state. This is done in a two-step recursive algorithm that uses the *a priori* PDF of the state. In most cases these calculations are intractable. However, for assumed Gaussian noise processes, the calculations reduce to computing multidimensional expectation integrals, or identically, multidimensional Gaussian-weighted integrals.

The classical filters that recursively compute these multidimensional integrals include the unscented Kalman filter (UKF),[2] the quadrature Kalman filter[3,4] otherwise known as the Gauss-Hermite filter (GHF), and the cubature Kalman filter (CKF).[5] All of these filters share the fact that they propagate a set of points and weights through a nonlinear function as a method to approximate multidimensional expectation integrals. The points and weights are carefully constructed in order to achieve the best possible accuracy. Some methods achieve this through *moment matching*, i.e., matching various orders of statistical moments of the underlying distribution. Other methods do this using numerically accurate *quadrature* or *cubature* rules, i.e., numerical integration methods. For instance, the UKF was designed from a moment matching perspective; however, it is also a special case of the CKF which uses a cubature rule. In the present work, we focus on quadrature and cubature-based filters, but include some state-of-the-art moment matching filters as well.

The GHF is constructed from a Cartesian product of Gauss-Hermite quadrature points and weights.[3] Although the GHF is very accurate, it is susceptible to the curse of dimensionality, and thus the sparse grid GHF[6] was developed in an attempt to overcome it. The CKF applies a third-degree spherical-radial cubature rule to

Further author information: (Send correspondence to Q.B.)
Q.B.: E-mail: butlerq@mcmaster.ca
S.A.G.: E-mail: gadsden@mcmaster.ca

Signal Processing, Sensor/Information Fusion, and Target Recognition XXXIII, edited by Ivan Kadar, Erik P. Blasch, Lynne L. Grewe, Proc. of SPIE Vol. 13057, 1305704 · © 2024 SPIE · 0277-786X · doi: 10.1117/12.3013683

compute the expectation integrals with a minimal number of evaluation points.[5] In fact, the cubature rule can be seen as a combination of a cubature rule and a first-degree (single point) quadrature rule. The high-degree CKF (HDCKF) uses a fifth-degree cubature rule to attain higher accuracy at the cost of more evaluation points, but also opens the possibility of negative weights.[7] The cubature-quadrature Kalman filter (CQKF) generalizes the CKF by applying the same cubature rule, but extends the quadrature rule to an arbitrary degree.[8] Other generalization of the CKF include the seventh-degree CKF,[9] high-degree CQKF,[10] and the generalized cubature quadrature Kalman filter.[11] The lattice Kalman filter employs a quasi-Monte Carlo integration technique called lattice rules.[12] A filter based on the conjugate unscented transformation (CUT) applies a cubature rule designed to compute arbitrarily high moments of a Gaussian distribution using fewer points than the GHF. The CUT filter has been shown to achieve higher accuracy than a particle filter using 5000 sample points.[13]

Beyond achieving the best possible accuracy, filters must also be robust. This includes robustness to initial conditions, measurement outliers, model uncertainties, disturbances, etc. Robust filtering methods include the Sliding innovation filter (SIF),[14] adaptive SIF,[15] alpha SIF,[16] and the robust variant of the lattice Kalman filter, the sliding innovation lattice filter.[17] Robust filters allow for accurate state and parameter estimation in application such as cognitive dynamic systems,[18] target tracking,[19] fault detection for a magnetorheological damper,[20] and preload loss detection in a ball screw feed drive.[21]

In this work, a filter is proposed that uses a unique quadrature method known as double exponential (DE) quadrature. The theory of double exponential quadrature is briefly presented next. This is followed by an overview of Bayesian filtering and how DE quadrature can be used to compute multivariate Gaussian-weighted integrals. Afterwards, the proposed filtering algorithm is outlined. A benchmark example of a range and bearing target tracking simulation is presented. Lastly, a brief conclusion summarizes the findings of this work.

2. DOUBLE EXPONENTIAL QUADRATURE

Gaussian quadrature has received much attention and practical use in research, and is indeed an elementary subject in many introductory numerical methods textbooks too. However, one need only combine a variable transformation with the simple trapezoidal rule to arrive at a family of powerful quadrature formulae. These quadrature formulae are called double exponential quadratures and have several advantages including: robustness against endpoint singularities, double exponential decay of the integrand, geometric convergence of the approximation error, and, in a sense, are optimal.[22–24] In addition, the weights and points are easily evaluated. Before DE quadrature is presented, we briefly review quadrature, otherwise known as numerical integration.

A quadrature formula is used to calculated the integral of a function $f(x)$ of the form

$$I(f) = \int_{\Omega} \omega(x) f(x)\, dx, \tag{1}$$

where $\omega(x)$ is a weight function and Ω is the domain of integration. A quadrature formula I_N is an approximation to the integral $I(f)$ in (1),

$$I_N \approx \sum_{i=1}^{N} w_i f(x_i), \tag{2}$$

where x_i are the evaluation points (or nodes), w_i are the weights, and N is the total number of nodes. Gaussian quadrature takes the x_i to be the roots of certain orthogonal polynomials. Together the x_i and w_i are determined by solving a system of nonlinear equations called moment equations. See Ref. 25 for a classic text on the subject.

In the context of recursive estimation and filtering, we aim to solve multivariate expectation integrals of the type

$$\mathrm{E}\left[f(x)\right] = \int_{\mathbb{R}^n} f(x)\, \mathcal{N}\left(x; \mu, \Sigma\right)\, dx, \tag{3}$$

where $f(x)$ is a vector-valued function of the state vector x, the weight function is now a multivariate normal distribution with mean μ and covariance Σ represented by $\mathcal{N}\left(x; \mu, \Sigma\right)$, and $\mathbb{R}^n$ is an n-dimensional vector space. The integral (3) can be solved using a Cartesian product of quadrature formulae as in the GHF.[3] However, the number of function evaluations grows exponentially with the state dimension n. A more efficient solution is to use a cubature rule as in the CKF and its variants[5,7–11] or the CUT filter.[13]

The family of double exponential quadratures include formulae for finite domains, the half-infinite interval, and the infinite interval.[22,26] These methods were designed with the intention of integrating functions with endpoint singularities, i.e., the integrand is singular at one or both bounds of integration. This is achieved through a variable transformation that causes the integrand to decay at a "double exponential" rate in both directions of the real line. As such, the DE transformations map an infinite domain into the original domain of integration. DE quadrature sees most of its use in experimental mathematics to calculate quantities with extreme precision, and has been shown to outperform Gaussian quadrature in accuracy and runtime performance.[27] The method has also been extended to multiple integration[28] and in the computation of multivariate integrals with endpoint singularities.[29–31] A comparison of the distribution of sample points for DE quadrature and Gauss-Legendre quadrature over the interval $[-1, 1]$ is shown in Fig. 1. DE quadrature places more points towards the endpoints of the interval which allows for improved accuracy when faced against endpoint singularities.

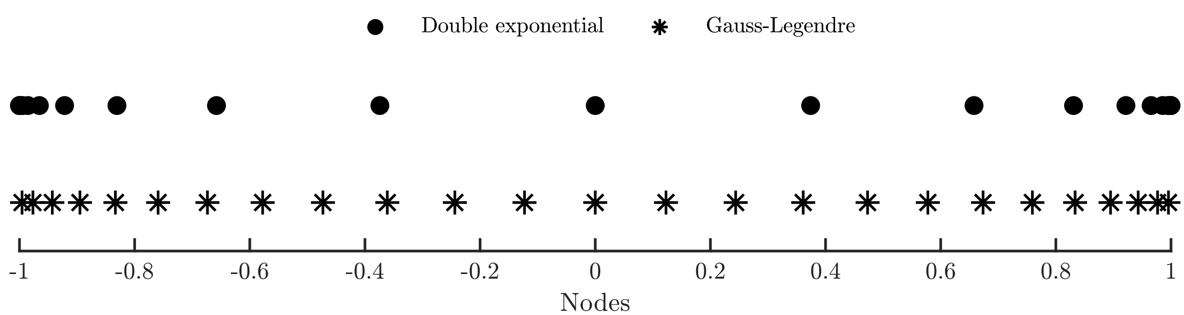

Figure 1. Distribution of the sample points (nodes) for the double exponential variable transformation and Gauss-Legendre quadrature over the interval $[-1, 1]$.

As is required by the proposed filter in Sec. 3, the domain of integration will be restricted to the half-infinite interval $\Omega = (0, \infty)$ and the weight function will be $\omega(x) = \exp(-x)$. Thus, we are approximating the integral

$$I(f) = \int_0^\infty f_1(x)e^{-x}\,dx. \tag{4}$$

DE quadrature maps the original domain of integration $(0, \infty)$ to the whole real line $(-\infty, \infty)$ through a variable substitution:

$$I(f) = \int_{-\infty}^{\infty} g(t)\,dt, \quad g(t) = f(\Psi(t))\Psi'(t), \tag{5}$$

where $f(x) = f_1(x)e^{-x}$ as in (4). The transformed integrand $g(t)$ decays double-exponentially over the real line. The trapezoidal rule is most efficient among quadratures with equidistant evaluation points for exponentially decaying integrands.[32–34] Thus, let the set of uniformly spaced points be denoted as

$$t = \{ih \,|\, i = \pm 1, 2, \ldots\}, \tag{6}$$

where h is the step size or spacing between the points. The DE transformation for the integral (4)[26] is

$$x_i = \Psi(t_i) = \exp(t_i - \exp(-t_i)) \tag{7}$$
$$\Psi'(t_i) = (1 + \exp(-t_i))\exp(t_i - \exp(-t_i)). \tag{8}$$

If we truncate t at $i = \pm n$, we get the DE formula

$$I_N = \sum_{i=-n}^{n} hf(\Psi(t_i))\Psi'(t_i), \tag{9}$$

and the total number of evaluation points is $N = 2n + 1$. An optimal step size can be selected as a function of n by equating the discretization error and truncation error of the quadrature method.[32,33] The step size is then

$$h(n) = \frac{2}{N}\,\mathrm{W}(\pi N), \tag{10}$$

where $\mathrm{W}(z)$ is the Lambert W-function which is an implicit solution to $z = we^w$ for w.

The DE quadrature method given by Equations (6), (7), and (8), and the optimal step size in Equation (10), will be used to derive the proposed filter.

3. FILTER FORMULATION

Consider a discrete, nonlinear dynamic system and measurement model with additive Gaussian noise given by the difference equations

$$\begin{aligned} x_k &= f(x_{k-1}) + w_{k-1} \\ y_k &= h(x_k) + v_k\,, \end{aligned} \tag{11}$$

where k is the discrete time step, $x_k \in \mathbb{R}^n$ is the state of the dynamic system, $y_k \in \mathbb{R}^m$ is the measurement, f is the process model, h is the measurement model, $w_{k-1} \in \mathbb{R}^n$ is Gaussian process noise with covariance Q_{k-1}, and $v_k \in \mathbb{R}^m$ is Gaussian measurement noise with covariance R_k. It is assumed that the noise vectors w_k and v_k are zero mean and uncorrelated.

3.1 Bayesian Filtering

In Bayesian filtering theory, the posterior density of the state gives a complete statistical description of the state. Upon receiving a measurement at time step k, the posterior density of the state at time $k-1$ is updated in two steps: the *time update* and the *measurement update*. For Gaussian noise processes, the update steps reduce to computing the means and covariances of both the predictive (prior) density and posterior density of the current state. These computations are in the form of Gaussian-weighted integrals over $\mathbb{R}^n$.

1. *Time Update*
   The mean of predictive density $\hat{x}_{k|k-1}$ and the error covariance $P_{k|k-1}$ are computed as

$$\begin{aligned} \hat{x}_{k|k-1} &= \mathrm{E}\left[f(x_{k-1})|y_{1:k-1}\right] \\ &= \int_{\mathbb{R}^n} f(x_{k-1})\,\mathcal{N}(x_{k-1};\hat{x}_{k-1|k-1},P_{k-1|k-1})dx_{k-1}, \end{aligned} \tag{12}$$

$$\begin{aligned} P_{k|k-1} &= \mathrm{E}\left[(x_k-\hat{x}_{k|k-1})(x_k-\hat{x}_{k|k-1})^T|y_{1:k-1}\right] \\ &= \int_{\mathbb{R}^n} f(x_{k-1})f^T(x_{k-1})\,\mathcal{N}(x_{k-1};\hat{x}_{k-1|k-1},P_{k-1|k-1})dx_{k-1} - \hat{x}_{k|k-1}\hat{x}^T_{k|k-1} + Q_{k-1}, \end{aligned} \tag{13}$$

   where $y_{1:k-1}$ is the sequence of measurements for times $k = 1, 2, \ldots, k-1$ and $\mathcal{N}(\cdot,\cdot)$ represents a Gaussian density.

2. *Measurement Update*
   The measurement likelihood density is given by

$$p(y_k|y_{1:k-1}) = \mathcal{N}\left(y_k;\hat{y}_{k|k-1},P_{yy,k|k-1}\right), \tag{14}$$

   where the predicted measurement and associated covariance are

$$\hat{y}_{k|k-1} = \int_{\mathbb{R}^n} h(x_k)\,\mathcal{N}\left(x_k,\hat{x}_{k|k-1},P_{k|k-1}\right)dx_k$$

$$P_{yy,k|k-1} = \int_{\mathbb{R}^n} h(x_k)h^T(x_k)\,\mathcal{N}\left(x_k;\hat{x}_{k|k-1},P_{k|k-1}\right)dx_k - \hat{y}_{k|k-1}\hat{y}^T_{k|k-1} + R_k. \tag{15}$$

The cross-covariance is calculated as

$$P_{xy,k|k-1} = \int_{\mathbb{R}^n} x_k h^T(x_k)\,\mathcal{N}\left(x_k; \hat{x}_{k|k-1}, P_{k|k-1}\right) dx_k - \hat{x}_{k|k-1}\hat{y}_{k|k-1}^T. \tag{16}$$

Upon receiving a measurement at time step k, the posterior density is given by

$$p(x_k|y_{1:k}) = \mathcal{N}\left(x_k; \hat{x}_{k|k}, P_{k|k}\right) \tag{17}$$

where

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \tag{18}$$

$$P_{k|k} = P_{k|k-1} - K_k P_{yy,k|k-1} K_k^T \tag{19}$$

$$K_k = P_{xy,k|k-1} P_{yy,k|k-1}^{-1}. \tag{20}$$

3.2 DE Quadrature for Multivariate Gaussian-Weighted Integrals

The Gaussian-weighted integrals given in Sec. 3.1 must be computed recursively. This can be done strictly using quadrature as in the GHF,[3] cubature as in the CKF,[5] high-degree CKF,[7] and CUT filter,[13] or as a combination of both cubature and quadrature as in the CQKF.[8] Here we extend the framework proposed by Ref. 8 to include double exponential quadrature instead of an arbitrarily high degree Gauss-Laguerre quadrature. The framework is restated up to the point of applying the quadrature rule.

Consider the Gaussian-weighted integral of an arbitrary function $f(x)$ where $x \in \mathbb{R}^n$,

$$I(f) = \int_{\mathbb{R}^n} f(x) \frac{1}{\sqrt{|\Sigma|(2\pi)^n}} \exp\left(-\tfrac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) dx, \tag{21}$$

where μ is the mean and Σ is the covariance. Using the variable substitution $x = \sqrt{\Sigma} r z + \mu$ the integral (21) can be expressed in spherical coordinates as

$$I(f) = \frac{1}{\sqrt{(2\pi)^n}} \int_0^\infty \int_{U_n} f(\sqrt{\Sigma} r z + \mu)\, d\sigma(z)\, r^{n-1} \exp\left(-\frac{r^2}{2}\right) dr, \tag{22}$$

where $\sqrt{\Sigma}$ is the matrix square root of the covariance, the norm of z is unity, i.e.,$||z|| = 1$, and U_n is the surface of the unit n-sphere, and $d\sigma(z)$ is the element surface area of U_n. The inner integral of (22) can be approximated using a third-degree spherical cubature rule

$$\int_{U_n} f(\sqrt{\Sigma} r z + \mu)\, d\sigma(z) \approx \frac{2\sqrt{\pi^n}}{2n\Gamma(\frac{n}{2})} \sum_{i=1}^{2n} f(\sqrt{\Sigma} r [u]_i + \mu), \tag{23}$$

where $[u]_i$ are the cubature points located at the intersections of the unit n-sphere with the coordinate axes. Substituting (23) into (22) yields

$$I(f) = \frac{1}{\sqrt{(2\pi)^n}} \int_0^\infty \left[\sum_{i=1}^{2n} \frac{2\sqrt{\pi^n}}{2n\Gamma(\frac{n}{2})} f(\sqrt{\Sigma} r [u]_i + \mu)\right] r^{n-1} \exp\left(-\frac{r^2}{2}\right) dr. \tag{24}$$

Using a second variable substitution, $r = \sqrt{2\Psi(t)}$, and after simplification, we get the desired form of the integral:

$$I(f) = \frac{1}{2n\Gamma(\frac{n}{2})} \int_0^\infty \left[\sum_{i=1}^{2n} f(\sqrt{2\Sigma}\sqrt{\Psi(t)}[u]_i + \mu)\right] \Psi(t)^{\frac{n}{2}-1} \exp(-\Psi(t))\, d\Psi(t). \tag{25}$$

Note that the integral in (25) is a one-dimensional integral over the half-infinite interval. Applying DE quadrature with points $\Psi(t) = \exp(t - \exp(-t))$ and weights $\Psi'(t) = (1 + \exp(-t))\exp(t - \exp(-t))$ with t as defined in (6) and h as in (10), we get the approximation

$$I(f) \approx \frac{1}{2n\Gamma(\frac{n}{2})} \sum_{j=1}^{N} \sum_{i=1}^{2n} h f\left(\sqrt{2\Sigma}\sqrt{\Psi(t_j)}[u]_i + \mu\right) \Psi'(t_j)\Psi(t_j)^{\frac{n}{2}-1} \exp(-\Psi(t_j)), \tag{26}$$

where we have indexed t from 1 to N, where N is the total number of quadrature points. Let us define the evaluation points ξ_i and associated weights w_i as

$$\xi_i = \sqrt{2\Psi(t_i)}[u]_i \tag{27}$$

$$\begin{aligned} w_i &= \frac{h}{2n\Gamma(\frac{n}{2})}\Psi'(t_i)\Psi(t_i)^{\frac{n}{2}-1}\exp(-\Psi(t_i)) \\ &= \frac{h}{2n\Gamma(\frac{n}{2})}\left(1+\exp(-t_i)\right)\left(\exp(t_i-\exp(-t_i))\right)^{\frac{n}{2}}\left(\exp(-\exp(t_i-\exp(-t_i)))\right). \end{aligned} \tag{28}$$

The integral approximation in (26) reduces to

$$I(f) \approx \sum_{i=1}^{2nN} w_i f\left(\sqrt{\Sigma}\xi_i + \mu\right). \tag{29}$$

3.3 The Cubature-DE Quadrature Kalman Filter

The filtering procedure is as follows:

1. *Initialization*:

$$\begin{aligned} \hat{x}_{0|0} &= \mathrm{E}\left[x_0\right] \\ P_{0|0} &= \mathrm{E}\left[(x_0-\hat{x}_0)(x_0-\hat{x}_0)^T\right] \end{aligned}$$

   Calculate evaluation points and weights according to (27) and (28).

2. *Time update*:

$$\begin{aligned} \mathcal{X}_{i,k-1|k-1} &= \sqrt{P_{k-1|k-1}}\xi_i + \hat{x}_{k-1|k-1} \\ \mathcal{X}^*_{i,k|k-1} &= f\left(\mathcal{X}_{i,k-1|k-1}\right) \\ \hat{x}_{k|k-1} &= \sum_{i=1}^{N} w_i \mathcal{X}^*_{i,k|k-1} \\ P_{k|k-1} &= \sum_{i=1}^{N} w_i \left[\mathcal{X}^*_{i,k|k-1} - \hat{x}_{k|k-1}\right]\left[\mathcal{X}^*_{i,k|k-1} - \hat{x}_{k|k-1}\right]^T + Q_{k-1} \end{aligned}$$

3. *Measurement update*:

$$\begin{aligned} \mathcal{X}_{i,k|k-1} &= \sqrt{P_{k|k-1}}\xi_i + \hat{x}_{k|k-1} \\ \mathcal{Y}_{i,k|k-1} &= h\left(\mathcal{X}_{i,k|k-1}\right) \\ \hat{y}_{k|k-1} &= \sum_{i=1}^{N} w_i \mathcal{Y}_{i,k|k-1} \\ P_{yy,k|k-1} &= \sum_{i=1}^{N} w_i \left(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1}\right)\left(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1}\right)^T + R_k \\ P_{xy,k|k-1} &= \sum_{i=1}^{N} w_i \left(\mathcal{X}_{i,k|k-1} - \hat{x}_{k|k-1}\right)\left(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1}\right)^T \end{aligned}$$

$$\begin{aligned} K &= P_{xy,k|k-1}P^{-1}_{yy,k|k-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K\left(y_k - \hat{y}_{k|k-1}\right) \\ P_{k|k} &= P_{k|k-1} - KP_{yy,k|k-1}K^T \end{aligned}$$

Note that the proposed filter uses $2nN$ points, which is the same as the CQKF. This is effectively N times more points than the CKF. For $N = 1$, the CQKF collapses down exactly to the CKF, while the DE filter nearly achieves the same result. Fig. 2 shows the number of evaluation points as a function of the state dimension for various filters. Notice that the DE filter may require more points for a lower state dimension ($n < 5$), but the number of points can be arbitrarily selected. In Sec. 4, the DE filter will be compared to the other filters with approximately the same number of points, and then compared with fewer points.

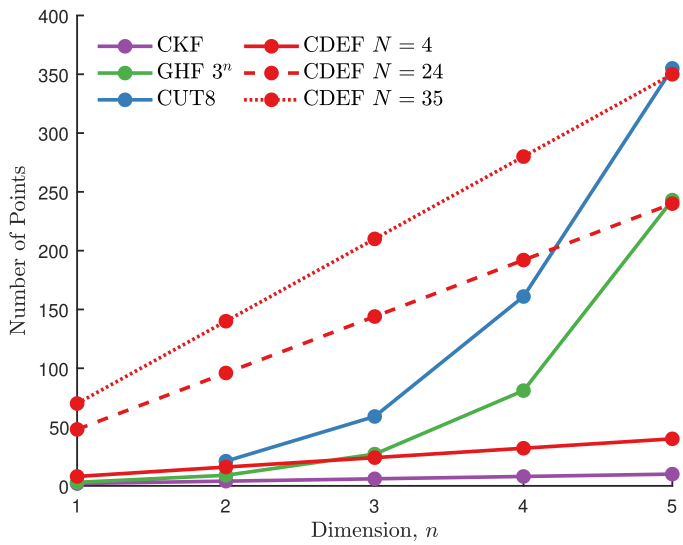

Figure 2. Number of evaluation points as a function of the state dimension n for the CKF, GHF with 3 points per dimension, CUT8, and the DE filter with $N = 4, 24$, and 35 quadrature points. Note the proposed DE filter is denoted as CDEF.

From a numerical perspective, the weights of the DE filter are all positive which leads to better numerical stability of the method. Let us use the notion of stability factor from Refs. 35 and 36 (see also Ref. 25) and defined as $\sum_i |w_i|$ (the sum of the absolute values of the weights). The stability factor gives insight into the roundoff errors introduced by the numerical method. It is desirable to have $\sum_i |w_i| = 1$ as when $\sum_i |w_i| \gg 1$ a large amount of roundoff error is introduced.[36] This can occur when some of the weights are negative. It is not rigorously proven here, but the proposed filter indeed achieves a stability factor of 1 for N large for an arbitrary state dimension as shown in Fig. 3. It is seen that for a smaller state dimension, the faster the stability factor reaches 1. This is an improvement over the UKF and the fifth- and seventh-degree CKFs. The third-degree CKF, GHF, and CUT8 filter achieve this result as well.

DE quadrature cannot be compared to other quadratures (such as Gaussian quadrature) based on the degree of exactness as it is not exact for any degree polynomial. Instead, DE quadrature, and other related quadratures, are compared by convergence analysis using the Euler-Mclaurin summation formula. In this sense, the DE quadrature formulae are exponentially convergent as they are built on top of the trapezoidal rule.[33]

4. SIMULATIONS AND RESULTS

4.1 Comparison to Similar Filters

In the following tracking problem we compare the performance of the proposed filter to several classical and state-of-the-art Gaussian filters. We denote the proposed cubature-DE quadrature Kalman filter as the CDEF. The classical filters include the CKF and UKF which are minimal and near minimal-point filters, respectively. The GHF can be considered best-in-class as it is extremely accurate since an arbitrary number of quadrature points can be used. However, it suffers from the curse of dimensionality. In this work, we compare to the GHF with three points per dimension, thus the points scale exponentially as 3^n (Fig. 2). The HDCKF was

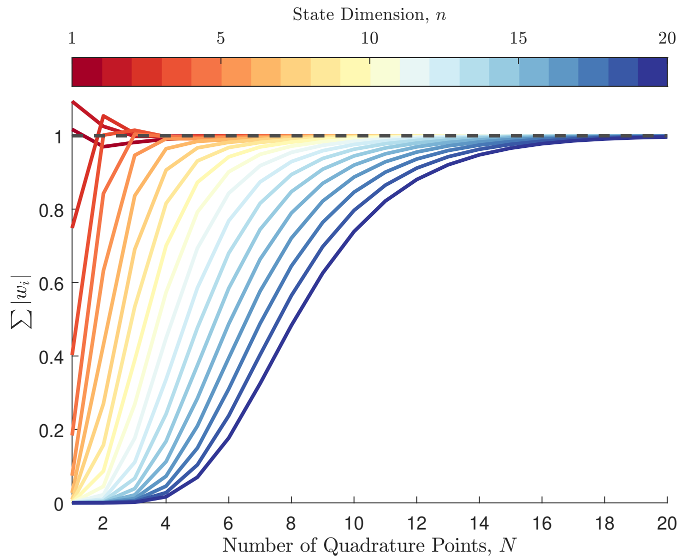

Figure 3. Stability factor for the proposed filter as a function of quadrature points N and state dimension n.

designed to compete against the GHF using far fewer points.[6] The HDCKF uses a fifth-degree cubature rule over the third-degree rule used in the CKF. Although this rule allows for negative weights which diminishes its numerical properties, much like negative weights in the UKF can cause a non-positive definite covariance matrix. The CQKF is most similar to the CDEF. It uses an arbitrary degree Guass-Laguerre quadrature for the radial integral (24). We compare the CDEF with an equal number of points as the CQKF when applicable. The conjugate unscented transform was first presented as a method to evaluate multidimensional expectation integrals.[13] The CUT consists of non-product cubature rules with positive weights that are determined through numerical optimization routines (depending on the desired order). The points and weights are chosen such that they exactly match predetermined moments of a Gaussian distribution, and can exactly integrate polynomial functions of the same order with respect to Gaussian probability density functions. Here we compare to the CUT8 which can exactly match up to the eighth moment of a Gaussian PDF by exploiting the symmetry of the distribution.

4.2 Simulations

The performance of the CDEF is compared in a range and bearing tracking problem. The air traffic control (ATC) scenario (and variants of the problem) has become a benchmark problem[7,13,37–39] and was adapted from the textbook example in Ref. 40. The ATC tracking scenario is as follows: A radar stationed at $[\xi, \eta] = [0\,\mathrm{m}, 0\,\mathrm{m}]$ provides range and bearing measurements of an aircraft flying by. The interval between measurements is $T = 5\,\mathrm{s}$. An aircraft, starting from $[\xi, \eta] = [25,000\,\mathrm{m}, 10,000\,\mathrm{m}]$ at time $t = 0\,\mathrm{s}$, flies westward for 125 s at 120 m/s before executing a 1°/s coordinated turn for 90 s. It then flies southward for 125 s at 120 m/s, followed by a 3°/s coordinated turn for 30 s. After the turn, it continues to fly westward at a constant velocity of 120 m/s. An example of the flight path and radar measurements (converted to Cartesian coordinates) are shown in Fig. 4.

The filters will use the coordinated turn model given in Eq. (30) with the state vector $x = [\xi, \dot{\xi}, \eta, \dot{\eta}, \Omega]^T$ where (ξ, η) is the position in XY coordinates, $(\dot{\xi}, \dot{\eta})$ is the velocity, and Ω is the turn rate, and w_{k-1} is the process noise. The measurements include the range and bearing as calculated in Eq. (31) where v_k is the measurement

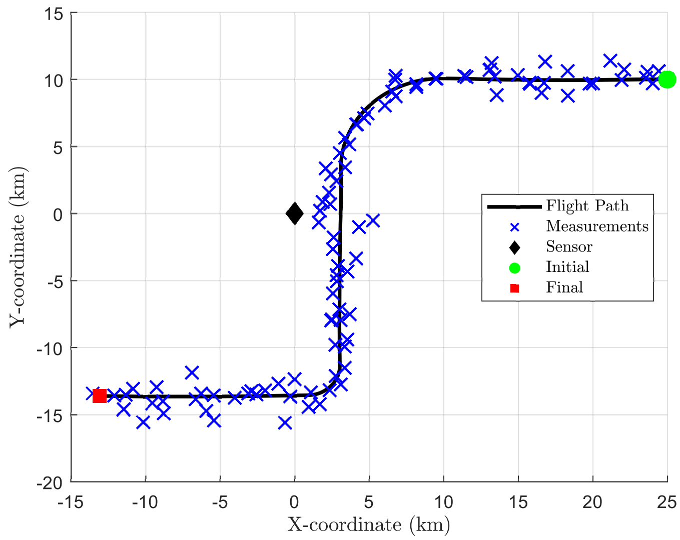

Figure 4. Example air traffic control scenario the true state (flight path) and measurements shown.

noise.

$$x_k = \begin{bmatrix} 1 & \dfrac{\sin(\Omega T)}{\Omega} & 0 & -\dfrac{1-\cos(\Omega T)}{\Omega} & 0 \\ 0 & \cos(\Omega T) & 0 & -\sin(\Omega T) & 0 \\ 0 & \dfrac{1-\cos(\Omega T)}{\Omega} & 1 & \dfrac{\sin(\Omega T)}{\Omega} & 0 \\ 0 & \sin(\Omega T) & 0 & \cos(\Omega T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + w_{k-1} \tag{30}$$

$$y_k = \begin{bmatrix} r_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \sqrt{(\xi_k)^2 + (\eta_k)^2} \\ \arctan\left(\dfrac{\eta_k}{\xi_k}\right) \end{bmatrix} + v_k \tag{31}$$

The process noise w_{k-1} and measurement noise v_k are zero-mean and Gaussian with covariance matrices

$$Q_{k-1} = L_1 \begin{bmatrix} \dfrac{T^3}{3} & \dfrac{T^2}{2} & 0 & 0 & 0 \\ \dfrac{T^2}{2} & T & 0 & 0 & 0 \\ 0 & 0 & \dfrac{T^3}{3} & \dfrac{T^2}{2} & 0 \\ 0 & 0 & \dfrac{T^2}{2} & T & 0 \\ 0 & 0 & 0 & 0 & \dfrac{L_2}{L_1}T \end{bmatrix} \quad \text{and} \quad R_k = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}. \tag{32}$$

The parameters used are $T = 5\,\mathrm{s}$, $L_1 = 0.16$, $L_2 = 0.01$, $\sigma_r = 100\,\mathrm{m}$, $\sigma_\theta = 1°$. The initial state estimate x_0 and initial state error covariance $P_{0|0}$ are

$$x_0 = [25,000\,\mathrm{m},\ -120\,\mathrm{m/s},\ 20,000\,\mathrm{m},\ 0\,\mathrm{m/s},\ 10^{-6}\,\mathrm{rad/s}]^T, \text{ and}$$
$$P_{0|0} = \mathrm{diag}([1000^2\,\mathrm{m}^2,\ 100\,\mathrm{m}^2/\mathrm{s}^2,\ 1000^2\,\mathrm{m}^2,\ 100\,\mathrm{m}^2/\mathrm{s}^2,\ (\pi/180)^2\,\mathrm{rad}^2/\mathrm{s}^2]).$$

For this 5-dimensional tracking problem, the following filters and their number of points were used: UKF with 11 points, CKF with 10 points, HDCKF with 53 points, GHF with 243 points, CQKF with 350 points, CUT8 filter with 355 points, and the CDEF with 350 points (or $N = 35$ quadrature points). Note that the tuning variable for the UKF was set to $\kappa = 1$ to avoid negative weights.

A simulation consisted of combining results over multiple Monte Carlo runs through the root-mean-squared errors defined below Sec. 4.3. Each simulation run consisted of initializing each filter with the same initial state estimate and state error covariance. Measurements were generated from the true trajectory and sensor model in Eq. (31). The same set of measurements were used for all filters for a particular run. The number of Monte Carlo runs was $N_R = 500$.

4.3 Results

All filters are compared using the root-mean-square error (RMSE) calculated at each time step for $N_R = 500$ Monte Carlo runs. The RMSEs for position, velocity, and turn rate are calculated as

$$\mathrm{RMSE}_{\mathrm{pos}}(k) = \sqrt{\frac{1}{N_R}\sum_{i=1}^{N_R}\Big((\xi_i(k) - \hat{\xi}_k(k))^2 + (\eta_i(k) - \hat{\eta}_k(k))^2\Big)},$$

$$\mathrm{RMSE}_{\mathrm{vel}}(k) = \sqrt{\frac{1}{N_R}\sum_{i=1}^{N_R}\Big((\dot{\xi}_i(k) - \dot{\hat{\xi}}_k(k))^2 + (\dot{\eta}_i(k) - \dot{\hat{\eta}}_k(k))^2\Big)},$$

$$\mathrm{RMSE}_{\Omega} = \sqrt{\frac{1}{N_R}\sum_{i=1}^{N_R}(\Omega_i(k) - \hat{\Omega}_i(k))^2},$$

where $\hat{x}_i(k) = [\hat{\xi}_i(k),\, \dot{\hat{\xi}}_i(k),\, \hat{\eta}_i(k),\, \dot{\hat{\eta}}_i(k),\, \hat{\Omega}_i(k)]^T$ is the filter estimate for run i at time step k, and $x_i(k) = [\xi_i(k),\, \dot{\xi}_i(k),\, \eta_i(k),\, \dot{\eta}_i(k),\, \Omega_i(k)]^T$ is the true state.

In Fig. 5 the RMSE for all filters at each time step are shown. It is interesting to note that all filter state errors attain a relatively constant level of error, but eventually begin to increase after some time. This is most notable for the CKF position RMSE which increases before $t = 50$ s, and for the UKF and CKF velocity RMSE which dramatically increase after $t = 50$ s. The UKF position RMSE maintains a comparable level to the GHF and CUT8 until about $t = 180$ s and then begins to diverge. The HDCKF, GHF, and CUT8 maintain a steady error for a longer period of time. The CUT8 maintains this for the longest time before diverging. Also, the CUT8 turn rate RMSE decreases initially before increasing to the same level as that of the GHF. The HDCKF under performed compared to the GHF, although the velocity RMSE of the two filters began to coincide after $t = 250$ s. The CQKF and the CDEF achieve the best overall RMSE over all time steps. Other than the CQKF turn rate error for $t > 350$ s (and perhaps the tail end of the velocity RMSE), the RMSE of these filters maintains a constant and low level of error.

For a fair comparison to the GHF which used 243 points, the number of points for the CDEF was reduced to 240. The simulation results for the GHF, CUT8 (with 355 points), and CDEF at the reduced number of points are shown in Fig. 6. Again, $N_R = 500$ Monte Carlo runs were used for the simulation. The CDEF still achieves better RMSE for all states than the GHF and the CUT8, despite the decrease in points. However, the CDEF begins to diverge to a higher level of error like the other filters after some time.

5. CONCLUSIONS

Nonlinear Gaussian filtering is approached using double exponential quadrature; a method unlike that of Gaussian quadrature which is typically used. The proposed CDEF out performs several state-of-the-art filters, even with fewer points. The performance achievement of the CDEF with much fewer points than the CUT8 shows that more complex cubature rules may not always be the best direction for computational efficiency and accuracy. It may be more worthwhile to investigate the application of more unique quadrature rules in nonlinear Gaussian filtering.

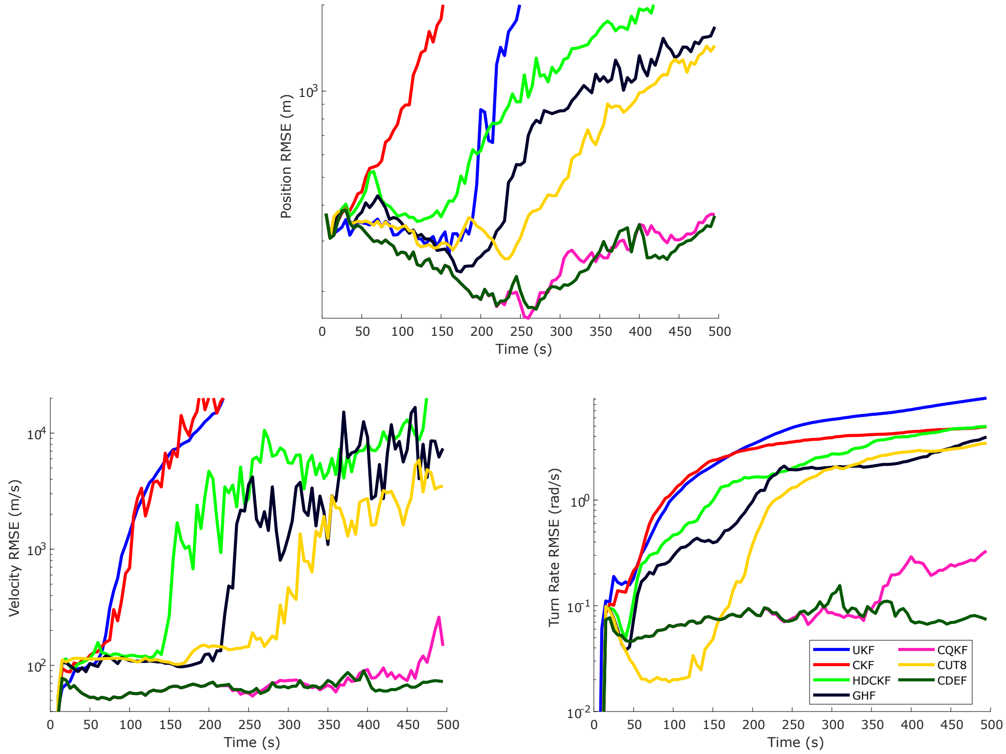

Figure 5. Position, velocity, and turn rate RMSE over 500 Monte Carlo runs for all filters.

REFERENCES

[1] Afshari, H. H., Gadsden, S. A., and Habibi, S., "Gaussian filters for parameter and state estimation: A general review of theory and recent trends," *Signal Processing* **135**, 218–238 (2017).

[2] Julier, S. and Uhlmann, J., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE* **92**, 401–422 (Mar. 2004).

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

[4] Arasaratnam, I., Haykin, S., and Elliott, R. J., "Discrete-Time Nonlinear Filtering Algorithms Using Gauss–Hermite Quadrature," *Proceedings of the IEEE* **95**, 953–977 (May 2007).

[5] Arasaratnam, I. and Haykin, S., "Cubature Kalman Filters," *IEEE Transactions on Automatic Control* **54**, 1254–1269 (June 2009).

[6] Jia, B., Xin, M., and Cheng, Y., "Sparse-grid quadrature nonlinear filtering," *Automatica* **48**, 327–341 (Feb. 2012).

[7] Jia, B., Xin, M., and Cheng, Y., "High-degree cubature Kalman filter," *Automatica* **49**, 510–518 (Feb. 2013).

[8] Bhaumik, S. and Swati, "Cubature quadrature Kalman filter," *IET Signal Processing* **7**(7), 533–541 (2013).

[9] Meng, D., Miao, L., Shao, H., and Shen, J., "A Seventh-Degree Cubature Kalman Filter," *Asian Journal of Control* **20**(1), 250–262 (2018).

[10] Singh, A. K. and Bhaumik, S., "Higher degree cubature quadrature kalman filter," *International Journal of Control, Automation and Systems* **13**, 1097–1105 (Oct. 2015).

[11] Wang, H., Zhang, W., Zuo, J., and Wang, H., "Generalized cubature quadrature Kalman filters: Derivations and extensions," *Journal of Systems Engineering and Electronics* **28**, 556–562 (June 2017).

[12] Rahimnejad, A., Gadsden, S. A., and Al-Shabi, M., "Lattice kalman filters," *IEEE Signal Processing Letters* **28**, 1355–1359 (2021).

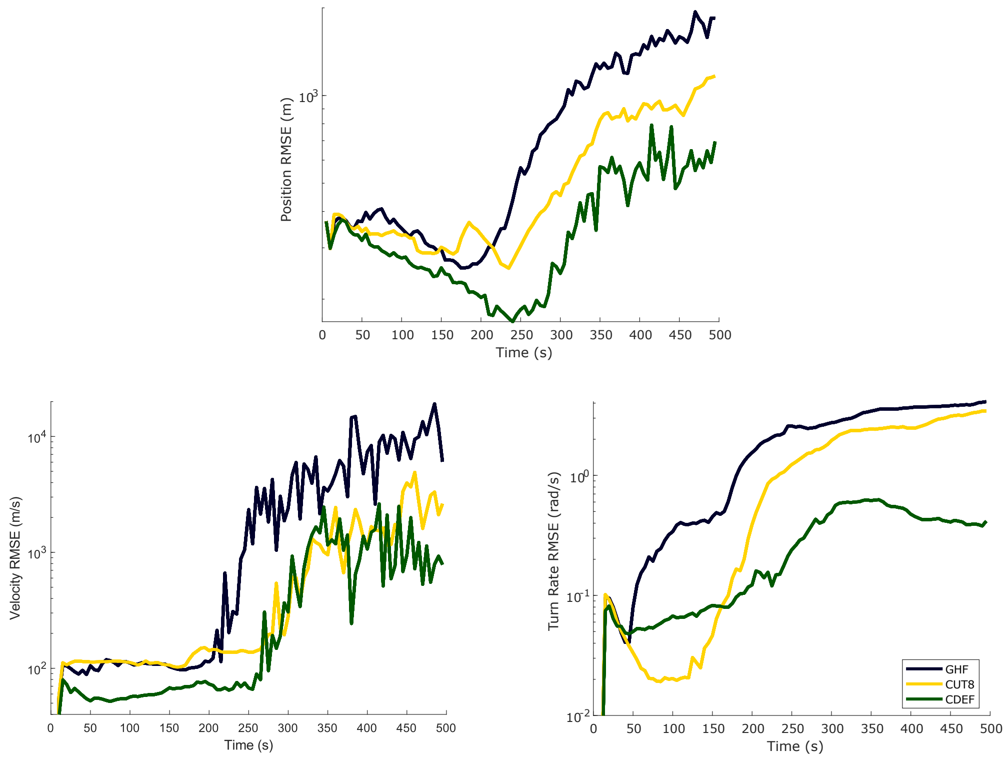

Figure 6. Position, velocity, and turn rate RMSE over 500 Monte Carlo runs for the GHF with 243 points, CUT8 with 355 points, and the CDEF with 240 points.

[13] Adurthi, N., Singla, P., and Singh, T., "Conjugate Unscented Transformation: Applications to Estimation and Control," *Journal of Dynamic Systems, Measurement, and Control* **140** (Nov. 2017).

[14] Gadsden, S. A. and Al-Shabi, M., "The sliding innovation filter," *IEEE Access* **8**, 96129–96138 (2020).

[15] Lee, A. S., Hilal, W., Gadsden, S. A., and Al-Shabi, M., "Combined kalman and sliding innovation filtering: An adaptive estimation strategy," *Measurement* **218**, 113228 (2023).

[16] AlShabi, M. and Gadsden, S. A., "Formulation of the alpha sliding innovation filter: A robust linear estimation strategy," *Sensors* **22**(22), 8927 (2022).

[17] Rahimnejad, A., Enayati, J., Vanfretti, L., Gadsden, S. A., and AlShabi, M., "Reinforced lattice kalman filters: A robust nonlinear estimation strategy," *IEEE Open Journal of Signal Processing* **4**, 410–423 (2023).

[18] Hilal, W., Gadsden, S. A., and Yawney, J., "Cognitive dynamic systems: A review of theory, applications, and recent advances," *Proceedings of the IEEE* **111**(6), 575–622 (2023).

[19] Edrisi, S., Enayati, J., Rahimnejad, A., and Gadsden, S. A., "A monte carlo-based iterative extended kalman filter for bearings-only tracking of sea targets," *Sensors* **24**(7), 2087 (2024).

[20] Lee, A. S., Wu, Y., Gadsden, S. A., and AlShabi, M., "Interacting multiple model estimators for fault detection in a magnetorheological damper," *Sensors* **24**(1), 251 (2023).

[21] Sicard, B. S., Butler, Q., Ziada, Y., Hughey, E., and Gadsden, S. A., "Preload loss detection in a ball screw system using interacting models," *IEEE Open Journal of Instrumentation and Measurement* **2**, 1–12 (2023).

[22] Takahasi, H. and Mori, M., "Double Exponential Formulas for Numerical Integration," *Publications of the Research Institute for Mathematical Sciences* **9**(3), 721–741 (1974).

[23] Mori, M., "Quadrature formulas obtained by variable transformation and the DE-rule," *Journal of Computational and Applied Mathematics* **12-13**, 119–130 (May 1985).

[24] Mori, M. and Sugihara, M., "The double-exponential transformation in numerical analysis," *Journal of Computational and Applied Mathematics* **127**, 287–296 (Jan. 2001).

[25] Davis, P. J. and Rabinowitz, P., [*Methods of numerical integration*], Courier Corporation (2007).

[26] Mori, M., "The Double Exponential Formulas for Numerical Integration over the Half Infinite Interval," in [*Numerical Mathematics Singapore 1988*], Agarwal, R. P., Chow, Y. M., and Wilson, S. J., eds., *International Series of Numerical Mathematics / Internationale Schriftenreihe zur Numerischen Mathematik / Série internationale d'Analyse numérique*, 367–379, Birkhäuser, Basel (1988).

[27] Bailey, D. H., Jeyabalan, K., and Li, X. S., "A Comparison of Three High-Precision Quadrature Schemes," *Experimental Mathematics* **14**, 317–329 (Jan. 2005).

[28] Muhammad, M. and Mori, M., "Numerical iterated integration based on the double exponential transformation," *Japan Journal of Industrial and Applied Mathematics* **22**, 77 (Feb. 2005).

[29] Mosig, J. R., "Weighted Averages and Double Exponential Algorithms," *IEICE Transactions on Communications* **E96.B**(10), 2355–2363 (2013).

[30] Michalski, K. A. and Mosig, J. R., "Efficient computation of Sommerfeld integral tails – methods and algorithms," *Journal of Electromagnetic Waves and Applications* **30**, 281–317 (Feb. 2016).

[31] Saye, R. I., "High-order quadrature on multi-component domains implicitly defined by multivariate polynomials," *Journal of Computational Physics* **448**, 110720 (Jan. 2022).

[32] Vanherck, J., Sorée, B., and Magnus, W., "Tanh-sinh quadrature for single and multiple integration using floating-point arithmetic," tech. rep. (July 2020). arXiv:2007.15057 [physics] type: article.

[33] Trefethen, L. N. and Weideman, J. A. C., "The Exponentially Convergent Trapezoidal Rule," *SIAM Review* **56**, 385–458 (Jan. 2014).

[34] Sugihara, M., "Optimality of the double exponential formula – functional analysis approach –," *Numerische Mathematik* **75**, 379–395 (Jan. 1997).

[35] Wu, Y., Hu, D., Wu, M., and Hu, X., "A Numerical-Integration Perspective on Gaussian Filters," *IEEE Transactions on Signal Processing* **54**, 2910–2921 (Aug. 2006).

[36] Stroud, A. H., [*Approximate Calculation of Multiple Integrals*], Prentice Hall, Englewood Cliffs, New Jersey (1971).

[37] Arasaratnam, I. and Haykin, S., "Square-Root Quadrature Kalman Filtering," *IEEE Transactions on Signal Processing* **56**, 2589–2593 (June 2008).

[38] Arasaratnam, I., Haykin, S., and Hurd, T. R., "Cubature Kalman Filtering for Continuous-Discrete Systems: Theory and Simulations," *IEEE Transactions on Signal Processing* **58**, 4977–4993 (Oct. 2010).

[39] Gadsden, A., Habibi, S., Dunne, D., and Kirubarajan, T., "Nonlinear Estimation Techniques Applied on Target Tracking Problems," *Journal of Dynamic Systems, Measurement, and Control* **134** (June 2012).

[40] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., [*Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*], John Wiley & Sons (Apr. 2002). Google-Books-ID: xz9nQ4wdXG4C.