Major Development Under Gaussian Filtering Since Unscented Kalman Filter

Abhinoy Kumar Singh, *Member, IEEE*

***Abstract*—Filtering is a recursive estimation of hidden states of a dynamic system from noisy measurements. Such problems appear in several branches of science and technology, ranging from target tracking to biomedical monitoring. A commonly practiced approach of filtering with nonlinear systems is Gaussian filtering. The early Gaussian filters used a derivative-based implementation, and suffered from several drawbacks, such as the smoothness requirements of system models and poor stability. A derivative-free numerical approximation-based Gaussian filter, named the unscented Kalman filter (UKF), was introduced in the nineties, which offered several advantages over the derivative-based Gaussian filters. Since the proposition of UKF, derivative-free Gaussian filtering has been a highly active research area. This paper reviews significant developments made under Gaussian filtering since the proposition of UKF. The review is particularly focused on three categories of developments: i) advancing the numerical approximation methods; ii) modifying the conventional Gaussian approach to further improve the filtering performance; and iii) constrained filtering to address the problem of discrete-time formulation of process dynamics. This review highlights the computational aspect of recent developments in all three categories. The performance of various filters are analyzed by simulating them with real-life target tracking problems.**

***Index Terms*—Bayesian framework, cubature rule-based filtering, Gaussian filters, Gaussian sum and square-root filtering, nonlinear filtering, quadrature rule-based filtering, unscented transformation.**

I. Introduction

THE modern era of science and technology has witnessed huge applications requiring state estimations from noisy measurements, e.g., target tracking, stochastic modeling, industrial diagnosis, and prognosis etc. [1]. Thanks to the filtering algorithms [1]–[5], which offer an efficient recursive state estimation tool. Filtering algorithms have helped in the modernization of several domains, such as space technology [1], medical [6], finance and economics [7], weather monitoring [8], etc.

The Bayesian framework [9], [10] is a common choice among the practitioners dealing with real-life estimation and filtering problems. It gives a probabilistic estimate of states by formulating prior and posterior probability density functions (pdf) [9], [10]. An analytical simplification is perceived by analyzing these pdfs numerically, and the Gaussian filters [1], [10] are most popular in this practice because of their high estimation accuracy at low computational cost. The Gaussian filters approximate the pdfs (prior and posterior) as Gaussian, and characterize them with mean and covariance.

Researchers and industrial practitioners engaged with estimation applications would be grateful to Rudolph E. Kalman, who developed an optimal state estimation and filtering technique [11] for linear systems with white Gaussian noises. In the filtering literature, this technique is popularly known as Kalman filter [1]–[4]. The Kalman filter can be implemented with several linearly approximated filtering problems in different domains, like in target tracking [12] and communication systems [13]. However, the linearity constraint restricts its application with many more practical problems, where the linear approximation gives a poor characterization of system dynamics. A nonlinear filter, named as extended Kalman filter (EKF) [1], [3], was developed in the latter half of the sixties. It introduced a derivative-based local linearization of system dynamics, and propagated the mean and covariance through the locally linearized system models. It was further modified to improve the filtering performance by introducing several variants [14]–[16]. However, the derivative-based local linearization introduces many disadvantages to the EKF and its variants, like the smoothness requirement for systems model, poor estimation accuracy, and low convergence rate [1]–[3]. Despite all these drawbacks, the EKF and its variants were the only alternative for more than three decades.

A derivative-free filtering method was introduced in the nineties, which is known as the unscented Kalman filter (UKF) [17]–[20]. The UKF approximates the desired pdfs as Gaussian, and characterizes them with mean and covariance. Although the Gaussian approximation of unknown pdfs encounters an error, it is more accurate compared to the EKF [17]–[20]. Furthermore, the mean and covariance are obtained from the first and second moments. The moment computation encounters an integral of the form "*nonlinear function* × *Gaussian distribution.*" Such integrals are generally intractable, and approximated with numerical methods. These methods approximate the integrals up to a particular order of Taylor series expansion. Thus, the moment of higher-order terms is another source of approximation error or the estimation error. Full expression of mean in terms of higher-order moments is derived in Appendix A. Nevertheless, it

Manuscript received October 12, 2019; revised February 4, 2020; accepted April 2, 2020. This work was supported by INSPIRE Faculty Award, Department of Science and Technology, Government of India. Recommended by Associate Editor Yebin Wang.

Citation: A. K. Singh, "Major development under Gaussian filtering since unscented Kalman filter," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 5, pp. 1308–1325, Sept. 2020.

A. K. Singh is with the Department of Electrical Engineering, Indian Institute of Technology Indore, Simrol, Indore 453552, India (e-mail: abhinoy.singh@iiti.ac.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2020.1003303

finds other sources of errors, such as noise Gaussian approximation and discrete-time formulation of process models, which appear for the EKF as well.

The UKF implements an unscented transformation-based numerical method of integral solving. Using the same numerical method, many variants of UKF [21]–[27] appear in the literature. The derivative-free filtering approach of UKF offers several advantages, such as better stability and improved estimation accuracy. Subsequently, the research continued further, and several derivative-free filtering methods were developed to further improve the performance. The aforementioned Gaussian filtering is a popular class of numerical approximation-based derivative-free filtering methods (particle filtering is another popular class of derivative-free filters; however, this review is particularly focused on the Gaussian filters).

The Gaussian filtering literature after the development of UKF can be classified in three categories: i) advancing numerical approximation accuracy; ii) modifying the Gaussian filtering approach against several limitations; and iii) removing the constraints on conventional Bayesian framework. This paper reviews the computational aspects of significant developments in all three categories.

In the first category, the numerical approximation accuracy of intractable integrals has been remarkably improved by updating the numerical approximation methods. The improved accuracy of numerical approximation of intractable integrals leads to accurate characterization of original pdfs, resulting in an enhanced estimation accuracy. The literature in this category can be further classified into two sub-categories: cubature rule-based filters and quadrature rule-based filters. Cubature rule-based filtering begins with cubature Kalman filter (CKF) [28], which is further modified with several variants [29]–[31]. Other important developments in this sub-category are cubature quadrature Kalman filter (CQKF) and its variants [32]–[34], high-degree CKF (HDCKF) [35] and high-degree CQKF (HDCQKF) [36]. The objective of sequentially developed filters is to improve the estimation accuracy. However, the computational burden is also increased with every development. Under the quadrature rule-based filtering, the literature begins with Gauss-Hermite filter (GHF) [37], [38] and its variants [39]–[41]. The estimation accuracy of GHF is better than the cubature rule-based filters. However, it failed to attract the practitioners due to its high computational burden. A series of further developments, namely the sparse-grid GHF (SGHF) [42]–[44], multiple GHF (MGHF) [45], [46], and adaptive SGHF (ASGHF) [47], appeared to reduce the computational burden. Another interesting development on improving the numerical approximation accuracy is transformed filtering [27], [34], [48], [49], where the sample points used for numerical approximation are orthogonally transformed. The transformed filtering could enhance the estimation accuracy without adding extra computational burden.

The second category reformulates the Gaussian filtering approach in different aspects. The square-root filtering [22], [33], [39], [50]–[52], and the Gaussian-sum filtering [30], [53]–[55] are popular developments in this category. The objective of square-root filtering is to preclude the positive-definite requirement of error covariance matrix (essentially required in traditional Gaussian filters). It does not impact the estimation accuracy, but extends the filter applicability to the systems where the positive-definiteness is not guaranteed. The Gaussian-sum filtering, on the other hand, improves the estimation accuracy by approximating the unknown pdfs with multiple Gaussian components instead of a single Gaussian.

In the third category, our discussion is focused on removing the constraint of discrete-time formulation of process dynamics. Under this constraint, the process dynamics is approximated as discrete, though it generally comes from continuous physical laws. In many problems (especially, if the sampling interval is large), this approximation is significant enough to harm the estimation accuracy. A continuous-discrete filtering technique [56]–[60] is developed to mitigate this approximation.

The literature witnesses other review papers pertaining to the Gaussian filtering, such as [61] and [62]. However, the scope of this review is widely different from them, more extensive, and more recent. Reference [61] was published in 2012, while many of the filtering techniques discussed in this review were introduced after that. Reference [61] discusses the computational aspects of the UKF only (out of all the filtering techniques reviewed in this paper), though a very brief discussion about the idea of CKF and GHF is also provided. It provides no discussion about square-root filtering, Gaussian-sum filtering, continuous-discrete filtering, and filtering with transformed sample points. The survey in [62] is focused more on the EKF and its extensions, though it puts some highlight on the UKF and the Gaussian-sum filtering.

The remaining part of this paper is organized as follows: Section II revisits the conventional Bayesian framework and the Gaussian filtering approach. A brief discussion about the UKF formulation is provided in Section III. Cubature rule-based filters are discussed in Section IV, followed by a brief discussion on quadrature rule-based filters in Section V. Section VI highlights the effect of orthogonal transformation over sample points. Sections VII and VIII describe square-root filtering and Gaussian-sum filtering, respectively. Section IX introduces the continuous-discrete filtering method. In Section X, the Gaussian filters are implemented with real-life nonlinear filtering problems, and their performance is analyzed with simulation results. Finally, discussions and conclusions are drawn in Section XI. A moment-based derivation of estimates is provided in Appendix A, followed by a brief discussion on Gaussian sigma point Kalman filtering algorithm in Appendix B.

II. Bayesian Framework of Filtering

The dynamic state space model of a system is represented with state and measurement equations, as [1]

$$\mathbf{x}_k = \phi_{k-1}(\mathbf{x}_{k-1}) + \boldsymbol{\eta}_k \tag{1}$$

$$\mathbf{y}_k = \gamma_k(\mathbf{x}_k) + \mathbf{v}_k \tag{2}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^d$ are state and measurement variables at the kth instant; $k \in \{0, 1, \ldots\}$, $\phi_k : \mathbf{x}_{k-1} \rightarrow \mathbf{x}_k$ and

$\gamma_k : \mathbf{x}_k \to \mathbf{y}_k$ are nonlinear functions, $\boldsymbol{\eta}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^d$ are process and measurement noises.

The filtering is a recursive estimation of state $\mathbf{x}_k$ on the receipt of a new measurement $\mathbf{y}_k$. The Bayesian framework is a conceptual solution, which offers a probabilistic interpretation. Therefore, the objective is to construct the pdf $P(\mathbf{x}_k|\mathbf{y}_{1:k})$ as soon as $\mathbf{y}_k$ is received. It is implemented in two steps: time update and measurement update, which are also known as prediction and update.

The time update step reveals how the conditional pdfs of $\mathbf{x}$ propagates in time. It is performed using Chapman-Kolmogorov equation, which gives [1], [62]

$$P(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int P(\mathbf{x}_k|\mathbf{x}_{k-1})P(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}. \tag{3}$$

$P(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is commonly known as the prior pdf. The measurement update corrects the prediction $P(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ using the noisy information received from the measurement $\mathbf{y}_k$. Subsequently, it constructs the pdf $P(\mathbf{x}_k|\mathbf{y}_{1:k})$ as [1], [62]

$$P(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{1}{c_k}P(\mathbf{y}_k|\mathbf{x}_k)P(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \tag{4}$$

where $P(\mathbf{y}_k|\mathbf{x}_k)$ is measurement likelihood obtained from (2) and c_k is a normalization constant, given as [1], [62]

$$c_k = \int P(\mathbf{y}_k|\mathbf{x}_k)P(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k. \tag{5}$$

$P(\mathbf{x}_k|\mathbf{y}_{1:k})$ is popularly known as the posterior pdf.

The remaining part of this paper represents $P(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ and $P(\mathbf{x}_k|\mathbf{y}_{1:k})$ as $P(\mathbf{x}_{k|k-1})$ and $P(\mathbf{x}_{k|k})$, respectively.

The Bayesian framework, being a probabilistic approach, fails to deliver a decisive solution. Therefore, a decisive solution is derived by analytical simplification of the pdfs. The Gaussian filtering is a commonly accepted solution due to its high estimation accuracy at low computational costs. The interest of this paper is limited to sigma point-based Gaussian filters, which is most commonly practiced in real-life. It encounters intractable integrals during the filtering, and approximates them numerically. The numerical approximation of integral is based on deterministically chosen sample points, known as sigma points, and associated weights. A brief discussion on sigma point-based Gaussian filters, also known as Gaussian sigma point Kalman filtering, is provided herewith.

In general, the Gaussian filtering is based on the following assumptions:

i) The prior and posterior pdfs are assumed to be Gaussian [62]–[64], i.e.,

$$P(\mathbf{x}_{k|k-1}) \approx \aleph(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{6}$$

$$P(\mathbf{x}_{k|k}) \approx \aleph(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \tag{7}$$

and

$$P(\mathbf{y}_{k|k-1}) \approx \aleph(\mathbf{y}_k; \hat{\mathbf{y}}_{k|k-1}, \mathbf{P}^{\mathbf{yy}}_{k|k-1}) \tag{8}$$

where $\aleph$ represents Gaussian distribution, $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{x}}_{k|k}$ are prior and posterior estimates of $\mathbf{x}_k$, $\mathbf{P}_{k|k-1}$ and $\mathbf{P}_{k|k}$ are prior and posterior covariances of $\mathbf{x}_k$, and, $\hat{\mathbf{y}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$ are estimate and covariance of $\mathbf{y}_k$.

ii) The noises, $\boldsymbol{\eta}_k$ and $\mathbf{v}_k$, are assumed to be uncorrelated, white and Gaussian with zero mean and covariance $\mathbf{Q}$ and $\mathbf{R}$, respectively. Thus, $E[\boldsymbol{\eta}_k] = E[\mathbf{v}_k] = E[\boldsymbol{\eta}_k\mathbf{v}_k^T] = 0$, $\boldsymbol{\eta}_k \sim \aleph(0,\mathbf{Q})$ and $\mathbf{v}_k \sim \aleph(0,\mathbf{R})$, where $\mathbf{Q} = E[\boldsymbol{\eta}_k\boldsymbol{\eta}_k^T]$ and $\mathbf{R} = E[\mathbf{v}_k\mathbf{v}_k^T]$.

The time update and measurement update steps for Gaussian sigma point Kalman filtering are discussed below.

1) Time Update

This step determines the prior estimate and covariance, i.e., $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, based on (3), which gives [1], [28], [38], [62]

$$\hat{\mathbf{x}}_{k|k-1} \approx \int \phi_{k-1}(\mathbf{x}_{k-1})\aleph(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})d\mathbf{x}_{k-1} \tag{9}$$

$$\begin{aligned}\mathbf{P}_{k|k-1} \approx \int & \phi_{k-1}(\mathbf{x}_{k-1})\phi_{k-1}(\mathbf{x}_{k-1})^T\aleph(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \\ & \mathbf{P}_{k-1|k-1})d\mathbf{x}_{k-1} - (\hat{\mathbf{x}}_{k-1|k-1})(\hat{\mathbf{x}}_{k-1|k-1})^T + \mathbf{Q}.\end{aligned} \tag{10}$$

2). Measurement Update

This step updates the prior estimate and covariance, $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, using the noisy information received from $\mathbf{y}_k$. The outcome is the posterior estimate and covariance, $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$, respectively. The computation of $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ requires the estimate and covariance of measurement ($\hat{\mathbf{y}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$), the cross-covariance between state and measurement ($\mathbf{P}^{\mathbf{xy}}_{k|k-1}$) and a Kalman gain ($\mathbf{K}_k$), which are computed as [1], [28], [38], [62]

$$\begin{aligned}\hat{\mathbf{y}}_{k|k-1} &= E[(\gamma_k(\mathbf{x}_k) + \mathbf{v}_k)|\mathbf{y}_{1:k-1}] \\ &\approx \int \gamma_k(\mathbf{x}_k)\aleph(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k\end{aligned} \tag{11}$$

$$\begin{aligned}\mathbf{P}^{\mathbf{yy}}_{k|k-1} &= E[(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})^T] \\ &\approx \int \gamma_k(\mathbf{x}_k)\gamma_k(\mathbf{x}_k)^T\aleph(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k \\ &\quad - (\hat{\mathbf{y}}_{k-1|k-1})(\hat{\mathbf{y}}_{k-1|k-1})^T + \mathbf{R}\end{aligned} \tag{12}$$

$$\begin{aligned}\mathbf{P}^{\mathbf{xy}}_{k|k-1} &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})^T] \\ &\approx \int \mathbf{x}_k\gamma_k(\mathbf{x}_k)^T\aleph(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k \\ &\quad - (\hat{\mathbf{x}}_{k|k-1})(\hat{\mathbf{y}}_{k|k-1})^T\end{aligned} \tag{13}$$

and

$$\mathbf{K}_k = \mathbf{P}^{\mathbf{xy}}_{k|k-1}(\mathbf{P}^{\mathbf{yy}}_{k|k-1})^{-1}. \tag{14}$$

Subsequently, $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ are obtained as [1], [28], [38], [62]

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \tag{15}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}^{\mathbf{yy}}_{k|k-1}\mathbf{K}_k^T. \tag{16}$$

A flow chart for implementing the Gaussian filtering approach is shown in Fig. 1.

The implementation of Gaussian filtering requires the solution of integrals in (9)–(13), which are mostly intractable for nonlinear systems. The Gaussian sigma point Kalman filtering approximates the integrals numerically using deterministically chosen sample points and associated weights. The integrals follow a general form of:

$$I(f) = \int f(\mathbf{x})\aleph(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})d\mathbf{x} \tag{17}$$

where $\mathbf{x}$ is a random variable with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$,

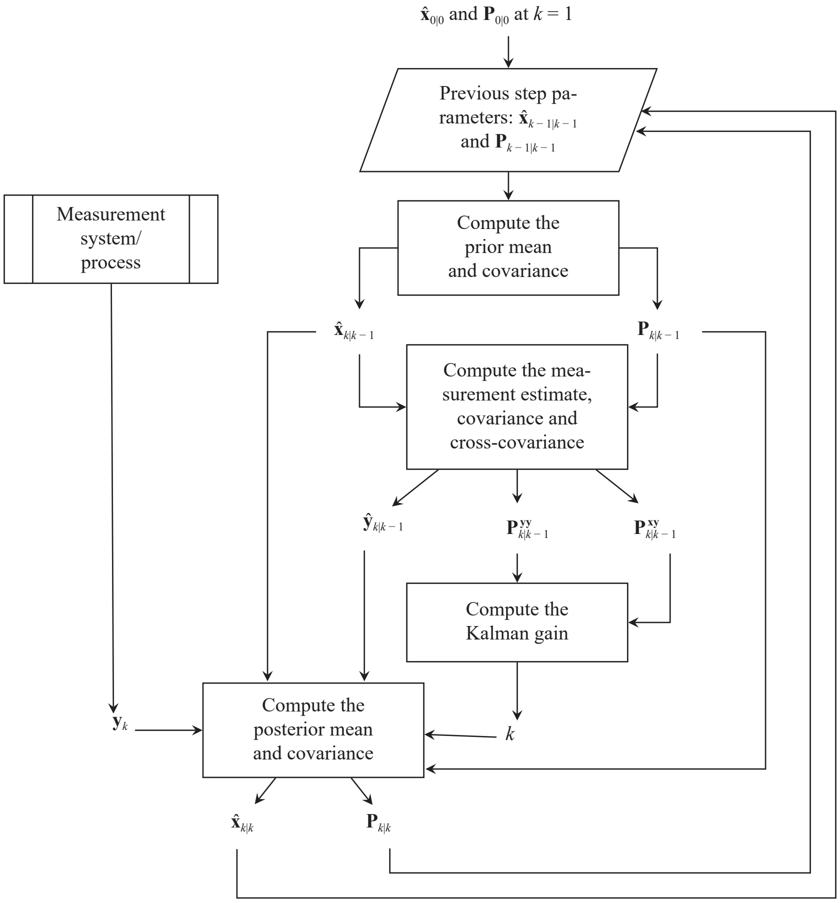

$\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$ at $k = 1$
Previous step parameters: $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$
Measurement system/ process
Compute the prior mean and covariance
$\hat{\mathbf{x}}_{k|k-1}$
$\mathbf{P}_{k|k-1}$
Compute the measurement estimate, covariance and cross-covariance
$\hat{\mathbf{y}}_{k|k-1}$
$\mathbf{P}^{\mathbf{yy}}_{k|k-1}$
$\mathbf{P}^{\mathbf{xy}}_{k|k-1}$
Compute the Kalman gain
$\mathbf{y}_k$
Compute the posterior mean and covariance
k
$\hat{\mathbf{x}}_{k|k}$
$\mathbf{P}_{k|k}$

Fig. 1. Block diagram for implementing ordinary Gaussian filters.

and $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a general nonlinear function of $\mathbf{x}$. The numerical methods used for integral approximation are generally defined for zero mean and unity covariance Gaussian, i.e., for $\aleph(\mathbf{x}; \mathbf{0}_{n\times 1}, \mathbf{I}_n)$ with $\mathbf{I}_n$ being a unit matrix and $\mathbf{0}_{n\times 1}$ being an n-dimensional array of all zero element. Let us denote this integral as $I_0(f)$, i.e.,

$$I_0(f) = \int f(\mathbf{x})\aleph(\mathbf{x}; \mathbf{0}_{n\times 1}, \mathbf{I}_n)d\mathbf{x}. \tag{18}$$

A numerical method, with $\boldsymbol{\xi}$ and $\mathbf{W}$ being the set of sample points and weights, respectively, approximates $I_0(f)$ as

$$I_0(f) \approx \sum_{i=1}^{N_s} \mathbf{W}_i f(\boldsymbol{\xi}_i) \tag{19}$$

where N_s is the number of sample points, and $\boldsymbol{\xi}_i$ and $\mathbf{W}_i$ are ith sample point and weight, respectively. The same numerical method can be extended for $\aleph(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})$ by simply transforming $\boldsymbol{\xi}$ with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$. Thus,

$$I(f) \approx \sum_{i=1}^{N_s} \mathbf{W}_i f(\hat{\mathbf{x}} + \boldsymbol{\Sigma}\boldsymbol{\xi}_i) \tag{20}$$

where $\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T = \mathbf{P}$. The algorithm for implementing the Gaussian sigma point Kalman filtering is presented in Appendix B.

In the remaining part of this survey, the Gaussian sigma point Kalman filtering is referred to as Gaussian filtering. However, the readers must take note that the term "Gaussian filtering" may have a broader meaning.

Section I discussed several Gaussian filters, broadly categorized in cubature rule-based filters and quadrature rule-based filters. These filters are different from each other mainly in terms of their numerical approximation technique. In fact, the implementation of the Gaussian filters are mostly identical, except for the sample points ($\boldsymbol{\xi}$) and associated weights ($\mathbf{W}$) used for approximating the intractable integrals.

Remark 1: Gaussian filters approximate unknown pdfs as Gaussian, which is not satisfied in most cases. Consequently, Gaussian filters are not optimal.

III. Unscented Kalman Filter

The UKF [17], [18] is the first nonlinear filter to introduce a derivative-free implementation in filtering with nonlinear systems. It utilizes an unscented transformation-based numerical approximation technique for approximating intractable integrals. The sample points generated using the unscented transformation are commonly known as sigma points. In its simplest form, the unscented transformation generates $2n + 1$ symmetrically distributed sigma points, as [17], [18]

$$\begin{aligned}
\boldsymbol{\xi}_{k|k}(0) &= \hat{\mathbf{x}}_{k|k} \\
\boldsymbol{\xi}_{k|k}(i) &= \hat{\mathbf{x}}_{k|k} + \left(\sqrt{(n+\kappa)\mathbf{P}_{k|k}}\right)_i \\
\boldsymbol{\xi}_{k|k}(n+i) &= \hat{\mathbf{x}}_{k|k} - \left(\sqrt{(n+\kappa)\mathbf{P}_{k|k}}\right)_i
\end{aligned} \tag{21}$$

where $i = 1, 2, \ldots, n$, $\boldsymbol{\xi}_{k|k}$ is the set of sigma points at kth instant, κ is constant (practitioner's choice) and $\left(\sqrt{(n+\kappa)\mathbf{P}_{k|k}}\right)_i$

represents the ith column of $\left(\sqrt{(n+\kappa)\mathbf{P}_{k|k}}\right)$. The preferred value of κ is $3-n$, i.e., $n+\kappa=3$, though it is not restricted. The weights are generated as [17], [18]

$$\begin{aligned}&\mathbf{W}(0)=\kappa/(n+\kappa)\\&\mathbf{W}(i)=\mathbf{W}(n+i)=1/(2(n+\kappa)).\end{aligned}\tag{22}$$

The UKF generates different set of sigma points in the time update ((9) and (10)) and measurement update ((11)–(16)); however, the associated weights are unchanged. The sigma points during the time update are generated using $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$. However, during the measurement update, they are generated using $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$. The expression of predicted mean for $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$ can be derived from Appendix A by substituting $\hat{\mathbf{x}}=\hat{\mathbf{x}}_{k-1|k-1}$ and $\sigma_{\mathbf{x}}=\left(\sqrt{(n+\kappa)\mathbf{P}_{k-1|k-1}}\right)$.

The UKF is further modified in scaled UKF [65] to improve the filtering performance. The scaled UKF reshapes the sigma points by introducing two scaling parameters, α and β, i.e., [65]

$$\begin{aligned}&\boldsymbol{\xi}_{k|k}(0)=\hat{\mathbf{x}}_{k|k}\\&\boldsymbol{\xi}_{k|k}(i)=\hat{\mathbf{x}}_{k|k}+\left(\sqrt{(n+\lambda)\mathbf{P}_{k|k}}\right)_i\\&\boldsymbol{\xi}_{k|k}(n+i)=\hat{\mathbf{x}}_{k|k}-\left(\sqrt{(n+\lambda)\mathbf{P}_{k|k}}\right)_i\end{aligned}\tag{23}$$

where $\lambda=\alpha^2(n+\kappa)-n$. The weights associated with the modified sigma points are reconstructed as [65]

$$\begin{aligned}&\mathbf{W}(0)=\lambda/(n+\lambda)\\&\mathbf{W}(i)=\lambda/(n+\lambda)+(1-\alpha^2+\beta)\\&\mathbf{W}(n+i)=1/(2(n+\lambda)).\end{aligned}\tag{24}$$

The parameter α $(0\le\alpha\le 1)$ controls the spread of the sigma points, while β reduces the higher-order errors discussed in Appendix A. In approximating a Gaussian pdf, as the case of Gaussian filtering, the optimal value of β is 2.

The unscented transformation approximates up to second-order Taylor series expansions; therefore, the higher-order terms contribute to approximation error.

The derivative-free implementation of UKF shows several advantages over the derivative-based EKF. Although both are approximated filtering approaches, the UKF is intended to offer a higher-order of approximation. Moreover, it precludes the smoothness requirement of system models. Note that the state dynamics may consist of sharp edges and lack of smoothness, which provide a large value for the derivative. Thus, the transition matrix may become inappropriately large and may cause poor accuracy or estimation failure for the EKF. In another drawback to the EKF, the measurement update step propagates the mean and covariance through locally linearized system models. The linearized models may be significantly mismatched with the true system model, especially if the nonlinearity is high, e.g., in high frequency oscillatory systems [66], [67]. The UKF precludes this drawback by propagating the mean and covariance through nonlinear functions representing the true system model. Furthermore, it should be mentioned that a second-order EKF [3], [14] is developed with a focus on improving the accuracy of EKF to match with UKF. However, it has high computational complexity and the derivative computation (which requires a smooth system model) remains a concern.

With all the advantages discussed above, the UKF attracts practitioners, and the same is apparent from a series of publications [68]–[71] available on its real-life implementation. Moreover, it is observed that the unscented transformation-based numerical approximation is not very accurate compared to the existing numerical methods of integral solving. This leaves a wide scope for further development. Many of the filtering methods discussed in this review aim to replace the unscented transformation method to improve numerical approximation accuracy, and thus, improve the estimation accuracy.

IV. Cubature Rule-Based Filters

The unscented transformation, used in UKF, is reportedly an inefficient numerical approximation technique given the existing literature on numerical approximation methods. An early replacement of unscented transformation is based on the spherical-cubature rule, which forms the basis for cubature rule-based filtering.

The cubature rule-based filters decompose the intractable integral $I(f)$ in a multivariate spherical integral and a univariate radial integral. The spherical integral is approximated using the spherical-cubature rule [72], [73], while the univariate radial integral is approximated using the quadrature rule [74], [75]. The literature witnesses a series of filters in this family by varying the order of spherical-cubature rule and quadrature rule.

Decomposition of Integral

Recalling (17), the intractable integral appears in the form of

$$\begin{aligned}I(f)&=\int f(\mathbf{x})\aleph(\mathbf{x};\hat{\mathbf{x}},\mathbf{P})d\mathbf{x}\\&=\frac{1}{\sqrt{|\mathbf{P}|(2\pi)^n}}\int_{\mathbb{R}^n}f(\mathbf{x})e^{-(1/2)(\mathbf{x}-\hat{\mathbf{x}})^T\mathbf{P}^{-1}(\mathbf{x}-\hat{\mathbf{x}})}d\mathbf{x}.\end{aligned}\tag{25}$$

The decomposition of $I(f)$ is based on the spherical-radial transformation of $\mathbf{x}\in\mathbb{R}^n$. The transformation is performed by substituting $\mathbf{x}=\boldsymbol{\Sigma}r\bar{\mathbf{Z}}+\hat{\mathbf{x}}$, where $r\in[0,\infty)$ is the radius and $\bar{\mathbf{Z}}$ is a unit vector representing direction. As $\bar{\mathbf{Z}}$ is a unit vector, $\bar{\mathbf{Z}}^T\bar{\mathbf{Z}}=1$ and $\mathbf{x}^T\mathbf{x}=r^2$. The decomposed intergral can be expressed as [28], [32], [35], [36]

$$I(f)=\frac{1}{\sqrt{(2\pi)^n}}\int_{r=0}^{\infty}\int_{U_n}\left[f(\boldsymbol{\Sigma}r\bar{\mathbf{Z}}+\hat{\mathbf{x}})d\sigma(\bar{\mathbf{Z}})\right]r^{n-1}e^{-r^2/2}dr\tag{26}$$

where U_n is surface of unit hyper-sphere and $d\sigma(\bar{\mathbf{Z}})$ is an elementary area over U_n.

The cubature rule-based filters are originally defined for zero mean and unity covariance Gaussian, i.e., for approximating $I_0(f)$. The decomposed $I_0(f)$ can be given as

$$I_0(f)\approx\frac{1}{\sqrt{(2\pi)^n}}\int_{r=0}^{\infty}\int_{U_n}\left[f(r\bar{\mathbf{Z}})d\sigma(\bar{\mathbf{Z}})\right]r^{n-1}e^{-r^2/2}dr.\tag{27}$$

The concerned spherical and radial integrals are

$$I_s=\int_{U_n}f(r\bar{\mathbf{Z}})d\sigma(\bar{\mathbf{Z}})\tag{28}$$

$$I_r=\int_{r=0}^{\infty}I_sr^{n-1}e^{-r^2/2}dr.\tag{29}$$

As discussed earlier, the same rule can be generalized for approximating $I(f)$ by transforming the sample points with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$.

A. Cubature Kalman Filter

The CKF is the earliest development under cubature rule-based filtering. It implements a third-degree spherical-cubature rule and first-order Gauss-Laguerre quadrature rule for approximating I_s and I_r, respectively.

1) Third-Degree Spherical-Cubature Rule

Using third-degree spherical-cubature rules, I_s could be approximated as [28]

$$I_s \approx \frac{2\sqrt{\pi^n}}{2n\Gamma(n/2)}\sum_{i=1}^{2n} f(r[\mathbf{u}]_i) \tag{30}$$

where Γ represents the gamma function and $[\mathbf{u}]$ is a set of indices representing the interstion points of unit hyper-sphere and coordinate axes, e.g., $[\mathbf{u}] = \{(1,0),\ (0,1),\ (-1,0), (0,-1)\}$ for a 2-dimensional system. The readers may refer to [28], [72], [73] for a detailed discussion on this rule.

2) First-Order Gauss-Laguerre Quadrature Rule

This rule states that an integral of form $\int_{\rho=0}^{\infty} f(\rho)\rho^{n-1}e^{-\rho^2}d\rho$ can be approximated as [28]

$$\int_{\rho=0}^{\infty} f(\rho)\rho^{n-1}e^{-\rho^2}d\rho \approx \omega_1 f(\rho_1) \tag{31}$$

where $\rho_1 = \sqrt{n/2}$ and $\omega_1 = \dfrac{\Gamma(n/2)}{2}$ are sample point and weight, respectively. The readers may refer to [28] for a detailed discussion.

3) Third-Degree Spherical-Radial Rule

The CKF combines the third-degree spherical-cubature rule with the first-order Gauss-Laguerre quadrature rule, and the combined rule is called the third-degree spherical-radial rule. To derive this rule, let us simplify $I_o(f)$ by substituting I_s from (30) into (27), and replacing $r = \sqrt{2}t$, i.e.,

$$I_o(f) \approx \frac{2}{2n\Gamma(n/2)}\int_{t=0}^{\infty}\left(\sum_{i=1}^{2n} f[\sqrt{2}t\mathbf{u}]_i\right)t^{n-1}e^{-t^2}dt. \tag{32}$$

This integral is in the form of (31). Thus, it can be approximated using the first-order Gauss-Laguerre quadrature rule, as [28]

$$I_o(f) \approx \frac{1}{2n}\left[\sum_{i=1}^{2n} f(\sqrt{n}[\mathbf{u}]_i)\right] \tag{33}$$

where $\sqrt{n}[\mathbf{u}]_i$ with $i \in \{1,2,\ldots,2n\}$ are sample points, named as cubature points.

As discussed earlier, (33) can be extended for approximating $I(f)$ as

$$I(f) \approx \frac{1}{2n}\sum_{j=1}^{2n} f(\hat{\mathbf{x}} + \mathbf{\Sigma}\sqrt{n}[\mathbf{u}]_j). \tag{34}$$

The following are some important literature on CKF highlighting its potential for practical applications:

i) Reference [76] studies the convergence of CKF.

ii) Reference [28] compares CKF with UKF, and studies its advantages over UKF.

iii) References [50], [77], and [78] discuss some applications of CKF in real-life problems.

The CKF is the same as the ordinary UKF with $\kappa = 0$. However, it is broadly accepted as it provides a specific choice of sample points and weights. Moreover, it introduces a new strategy for integral approximation, which combines the spherical and radial rules of integral approximation. This new strategy was of huge interest for researchers, leading to several extensions, such as HDCKF, CQKF, and HDCQKF. It will be shown in later discussions that the accuracy is improved with every extension, though the computational burden also increases.

B. High-Degree Cubature Kalman Filter

The HDCKF [35] implements a high-degree spherical-cubature rule [72], [79], [80] for approximating I_s, which was approximated with only third-degree spherical-cubature rule in the CKF. I_r is approximated using the same first-order Gauss-Laguerre quadrature rule used in CKF. The combined rule is known as the high-degree spherical-radial rule, and the resulting sample points are known as high-degree cubature points.

1) High-Degree Spherical-Cubature Rule

The high-degree spherical-cubature rule was developed by Genz [72] and utilized by Jia *et al.* [35] in the filtering domain. It is flexible in terms of degree (odd only) of approximation for intractable integrals. The discussion in this review is limited to its implementation in approximating I_s. The author refers to [72], [79], and [80] for a detailed discussion.

A $(2m_d + 1)$-degree spherical-cubature rule ($m_d \in \mathbb{Z}^+$, where $\mathbb{Z}^+$ is the set of positive integers) approximates I_s as [35]

$$I_s \approx \sum_i w_{\vartheta^i} f\{r\mathbf{u}_{\vartheta^i}\} \tag{35}$$

where $\vartheta^i = (\vartheta_1, \vartheta_2, \ldots, \vartheta_n)_i$, $\forall\ i \in \{1,2,\ldots,c_n\}$ with c_n being the number of possible combinations for which $\sum_{j=1}^n \vartheta_j = m_d$, while $(\vartheta_1, \vartheta_2, \ldots, \vartheta_n)_i$ is the ith combination of $(\vartheta_1, \vartheta_2, \ldots, \vartheta_n)$ satisfying $\sum_{j=1}^n \vartheta_j = m_d$. The set of high-degree cubature points and weights are obtained as

$$\{r\mathbf{u}_{\vartheta^i}\} \triangleq \{(\bar{\beta}_1 r\mathbf{u}_{\vartheta_1}, \bar{\beta}_2 r\mathbf{u}_{\vartheta_2}, \ldots, \bar{\beta}_n r\mathbf{u}_{\vartheta_n})\} \tag{36}$$

$$w_{\vartheta^i} \triangleq 2^{-(c_n - z_{\vartheta^i})}\left(\int_{U_n}\left(\prod_{j=1}^{n}\prod_{k=0}^{\vartheta_j - 1}\frac{\bar{\mathbf{Z}}_j^2 - \mathbf{u}_k^2}{\mathbf{u}_{\vartheta_j}^2 - \mathbf{u}_k^2}\right)d\sigma(\bar{\mathbf{Z}})\right) \tag{37}$$

where $\bar{\beta}_i = \pm 1$, z_{ϑ^i} is the number of zeros in ϑ^i, $\mathbf{u}_{\vartheta_j} = \sqrt{\vartheta_j/m_d}$, and $\bar{\mathbf{Z}}_j$ is a unit vector. In (37), the computation of w_{ϑ^i} encounters an integral of form $\int_{U_n}\bar{\mathbf{Z}}_1^{\delta_1}\bar{\mathbf{Z}}_2^{\delta_2}\ldots\bar{\mathbf{Z}}_n^{\delta_n}d\sigma(\mathbf{Z})$, where $\mathbf{Z} = [\bar{\mathbf{Z}}_1, \bar{\mathbf{Z}}_2, \ldots, \bar{\mathbf{Z}}_n]^T$. An approximated solution to such integrals is derived in [74] and [35] as

$$\int_{U_n}\bar{\mathbf{Z}}_1^{\delta_1}\bar{\mathbf{Z}}_2^{\delta_2}\ldots\bar{\mathbf{Z}}_n^{\delta_n}d\sigma(\bar{\mathbf{Z}}) \approx 2\frac{\Gamma((\delta_1+1)/2)\ldots\Gamma((\delta_n+1)/2)}{\Gamma((|\delta| + n)/2)}$$

where $|\delta| = \delta_1 + \delta_2 + \cdots + \delta_n$.

2) High-Degree Spherical-Radial Rule

To derive the high-degree spherical-radial rule, I_s is approximated using high-degree spherical-cubature rule ((35)), then substituted into (27) to simplify $I_0(f)$ in the form of a radial integral given as [35]

$$I_0(f) \approx \frac{1}{\sqrt{(2\pi)^n}} \int_{r=0}^{\infty} \Big[\sum_i w_{\vartheta i} f\{r\mathbf{u}_{\vartheta i}\} \Big] r^{n-1} e^{-r^2/2} dr. \tag{38}$$

Then, the radial integral is then approximated using the first-order Gauss-Laguerre quadrature rule ((31)), as [35]

$$I_0(f) \approx \frac{1}{2\sqrt{\pi^n}} \Big[\sum_i w_{\vartheta i} \omega_1 f\{\mathbf{u}_{\vartheta i}\rho_1\} \Big]$$

where $\mathbf{u}_{\vartheta i}\rho_1$ and $\frac{w_{\vartheta i}\omega_1}{2\sqrt{\pi^n}}$ are sample points and weights $\forall$ $i \in \{1,2,\ldots,c_n\}$.

Following the previous discussion, $I(f)$ can be approximated by transforming $\mathbf{u}_{\vartheta i}\rho_1$ with mean and covariance, i.e.,

$$I(f) \approx \frac{1}{2\sqrt{\pi^n}} \Big[\sum_i w_{\vartheta i} \omega_1 f\{\hat{\mathbf{x}} + \boldsymbol{\Sigma}\mathbf{u}_{\vartheta i}\rho_1\} \Big].$$

Note: $m_d = 1$ represents a third-degree spherical-cubature rule, hence HDCKF reduces to CKF under this condition.

The higher-degree approximation of I_s results in better estimation accuracy for the HDCKF compared to the CKF, but adds extra sample points, which also increases the computational burden. Note that a higher-order approximation is introduced only for I_s, and the order of approximation of I_r remains unchanged. Furthermore, the order of approximation of I_s is restricted to odd orders only. Subsequently, even a slight improvement in accuracy requires a two-order jump, requiring extra sample points.

C. Cubature Quadrature Kalman Filter

The CQKF [32] implements a higher-order Gauss-Laguerre quadrature rule for approximating I_r, instead of the first-order Gauss-Laguerre quadrature rule used in CKF and HDCKF. Moreover, it approximates I_s using the third-degree spherical-cubature rule utilized in CKF. The combined rule is named as cubature quadrature rule.

1) High-Order Gauss-Laguerre Quadrature Rule

An n'-order Gauss-Laguerre quadrature rule approximates the integral of form $\int_{\rho=0}^{\infty} f(\rho)\rho^{\upsilon}e^{-\rho}d\rho$ as [36]

$$\int_{\rho=0}^{\infty} f(\rho)\rho^{\upsilon}e^{-\rho}d\rho \approx \sum_{i'=1}^{n'} A_{i'} f(\rho_{i'}) \tag{39}$$

where υ is constant, while $\rho_{i'}$ and $A_{i'}$ $\forall$ $i' \in \{i,2,\ldots,n'\}$ are sample points and associated weights, respectively. The sample point $\rho_{i'}$ is given as i'th root of n'-order Chebyshev-Laguerre polynomial equation [74], [81]

$$L_{n'}^{\upsilon}(\rho) = (-1)^{n'} \rho^{-\upsilon} e^{\rho} \frac{d^{n'}}{d\rho^{n'}} \rho^{\upsilon+n'} e^{-\rho} = 0. \tag{40}$$

The weight $A_{i'}$ is obtained as

$$A_{i'} = \frac{n'!\Gamma(\upsilon + n' + 1)}{\rho_{i'}[\dot{L}_{n'}^{\upsilon}(\rho_{i'})]^2} \tag{41}$$

where $\dot{L}_{n'}^{\upsilon}(\rho_{i'})$ represents the first derivative of $L_{n'}^{\upsilon}(\rho)$ at $\rho = \rho_{i'}$.

The readers should note that $\rho_{i'}$ and $A_{i'}$ are designed for approximating the radial integral, not the desired integral $I(f)$ or $I_0(f)$. The author refers to [74], [81] for a detailed discussion on high-order Gauss-Laguerre quadrature rule.

2) Cubature Quadrature Rule

The derivation of the cubature quadrature rule is similar to the third-degree spherical-radial rule (Section IV-A-3)), except for $t = r^2/2$ instead of $t = r/\sqrt{2}$. The simplified $I_o(f)$ (with replacement of $t = r^2/2$) appears in the form of (39) with $v = n/2 - 1$. This integral is approximated using the n'-order Gauss-Laguerre quadrature rule, as [32]

$$I_0(f) \approx \frac{1}{2n\Gamma(n/2)} \left[\sum_{i=1}^{2n} \sum_{i'=1}^{n'} A_{i'} f(\sqrt{2\rho_{i'}}[\mathbf{u}]_i) \right]. \tag{42}$$

Following the previous discussion, $I(f)$ can be approximated as

$$I(f) \approx \frac{1}{2n\Gamma(n/2)} \left[\sum_{i=1}^{2n} \sum_{i'=1}^{n'} A_{i'} f\Big(\hat{\mathbf{x}} + \boldsymbol{\Sigma}(\sqrt{2\rho_{i'}}[\mathbf{u}]_i)\Big) \right]. \tag{43}$$

Note: $n' = 1$ represents the first-order Gauss-Laguerre quadrature rule, reducing the CKF to the CQKF under this condition.

The contribution of CQKF in cubature rule-based filtering is similar to the HDCKF, but it utilizes higher-order approximations for I_r instead of I_s. However, a major difference is that it allows any integer-order of approximation (for I_r), unlike the odd-order restriction (for I_s) in HDCKF. Similar to the HDCKF, it could enhance the estimation accuracy of the CKF with an increased computational burden.

D. High-Degree Cubature Quadrature Kalman Filter

The HDCQKF [36] is the most generalized form of cubature rule-based filtering, which reduces to each of the CKF, CQKF, and HDCKF under certain simplifications. It is the most accurate cubature rule-based filter, but also the computationally most inefficient. It approximates I_s using the high-degree spherical-cubature rule and I_r using the high-order Gauss-Laguerre quadrature rule. The combined rule is named as high-degree cubature quadrature rule.

1) High-Degree Cubature Quadrature Rule

The high-degree spherical-cubature rule simplifies $I_0(f)$ in the form of (38). Let us replace $r = \sqrt{2t}$ in this simplified expression, i.e.,

$$I_0(f) \approx \frac{1}{2\sqrt{\pi^n}} \int_{t=0}^{\infty} \left[\sum_i w_{\vartheta i} f\Big\{ \sqrt{2t}\mathbf{u}_{\vartheta i} \Big\} \right] t^{n/2-1} e^{-t} dt. \tag{44}$$

This integral is in the form of (39), hence $I_0(f)$ could be further approximated using the high-order Gauss-Laguerre quadrature rule, as [36]

$$I_0(f) \approx \frac{1}{2\sqrt{\pi^n}} \sum_{i'=1}^{n'} A_{i'} \left[\sum_i w_{\vartheta i} f\Big\{ \sqrt{2\rho_{i'}}\mathbf{u}_{\vartheta i} \Big\} \right]$$

where $\sqrt{2\rho_{i'}}\mathbf{u}_{\vartheta i}$ and $\frac{A_{i'}w_{\vartheta i}}{2\sqrt{\pi^n}}$ are sample points and associated weights with $i' \in \{1,2,\ldots,n'\}$ and $i \in \{1,2,\ldots,c_n\}$.

Following the previous discussion, the integral $I(f)$ can be approximated as

$$I(f) \approx \frac{1}{2\sqrt{\pi^n}} \sum_{i'=1}^{n'} A_{i'} \left[\sum_{|\vartheta|} w_\vartheta f\left\{ \hat{\mathbf{x}} + \mathbf{\Sigma} \sqrt{2\rho_{i'}} \mathbf{u}_\vartheta \right\} \right].$$

The following are some important notes about HDCQKF:

i) It is flexible in terms of degree of approximation for both I_s (by varying m_d) and I_r (by varying n').

ii) The estimation accuracy improves with increasing m_d as well as n'. However, the computational cost also increases.

iii) For $m_d = 1$ and $n' = 1$, it reduces to CKF.

iv) For $m_d \geq 2$ and $n' = 1$, it reduces to HDCKF.

v) For $m_d = 1$ and $n' \geq 2$, it reduces to CQKF.

The HDCQKF is competent for practical applications as it provides a fine trade-off between accuracy and computational burden by varying m_d and n'. It is a generalized filter in terms of order of approximation for I_s and I_r, except that the even-order of approximation is not possible for I_s.

It should be noted that the computational burden increases consistently with successive development in cubature rule-based filtering. However, the computational burden still increases with polynomial order as the system dimension increases. Subsequently, the cubature rule-based filters do not suffer from the *curse of conditionality* problem, as the case of the quadrature rule-based filters. Thus, the cubature rule-based filters can be implemented with relative higher dimension compared to the quadrature rule-based filters.

V. Quadrature Rule-Based Filters

The quadrature rule-based filtering is another popular category under the Gaussian filtering. The filters in this category are known for high estimation accuracy, but their practical applicability is limited due to high computational costs. However, there is ongoing research to reduce the computational burden, so that, the high estimation accuracy can be celebrated in a broader sense.

The filters in this category utilize Gauss-Hermite quadrature rule for approximating the intractable integrals. It is a univariate numerical approximation rule, while practical problems are mostly multivariate. Therefore, an additional mathematical law is implemented for extending the univariate rule in multivariate domain. The literature witnesses several filters with different approaches used for this extension. This section first introduces the univariate Gauss-Hermite quadrature rule, then reviews the existing filters.

A. Gauss-Hermite Quadrature Rule

Let us assume x be a univariate random variable. Then, the Gauss-Hermite quadrature rule can be represented as [38]

$$\int_{-\infty}^{\infty} f(x) \frac{1}{(2\pi)^{1/2}} e^{-x^2} dx \approx \sum_{i=1}^{N_{s1}} f(q_i)\omega_i \tag{45}$$

where q_i and ω_i $\forall$ $i \in \{1,2,\ldots,N_{s1}\}$ are univariate sample points and associated weights, respectively, and N_{s1} is the number of univariate sample points. Two popular methods for generating q_i and ω_i are the moment matching method [42] and Golub's method [38], [48]. The sample points q are commonly known as quadrature points [38], [42], [47].

1) Moment Matching Method

The moment matching method generates q_i and ω_i by matching N_{s1} moments. The moment equations are formulated as [10], [12]

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ q_1 & q_2 & \cdots & q_{N_{s1}} \\ \vdots & \vdots & \ddots & \vdots \\ q_1^{N_{s1}-1} & q_2^{N_{s1}-1} & \cdots & q_{N_{s1}}^{N_{s1}-1} \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{N_{s1}} \end{pmatrix} = \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_{N_{s1}-1} \end{pmatrix}$$

where $M_0, M_1, \ldots, M_{N_{s_1}-1}$ are moments of order 0 to $N_{s_1} - 1$.

The moment matching principle offers N_{s1} equations. However, the number of unknowns is $2N_{s1}$ (N_{s1} quadrature points and N_{s1} weights). Two approaches are commonly used to tackle this problem: i) quadrature points are chosen arbitrarily, then the weights are obtained by solving moment equations; and ii) quadrature points are determined as zeros of Hermite polynomial [82], [83], then the weights are obtained by solving moment equations.

2) Galub's Method

Golub's method is a simplification of the moment matching method itself. However, it is broadly used as an independent method [38], [46], [48] in filtering literature, being the reason for a separate discussion in this review. Its computational simplicity and specific choice of quadrature points and weights attracts practitioners. The desired quadrature points and weights are generated by formulating a symmetric tridiagonal matrix $\mathbf{J}$, i.e., $\mathbf{J}_{j,j} = 0$ $\forall$ $j \in \{1,2,\ldots,N_{s1}\}$ and $\mathbf{J}_{i,i+1} = \sqrt{i/2}$ $\forall$ $i \in \{1,2,\ldots,N_{s1}-1\}$. If Λ_i ($i \in \{1,2,\ldots,N_{s1}\}$) is the ith eigenvalue of $\mathbf{J}$, then ith quadrature point is $q_i = \sqrt{2}\Lambda_i$ [38], [48]. The weight ω_i is square of the first element of ith normalized eigenvector of $\mathbf{J}$.

B. Gauss-Hermite Filter

The GHF [37], [38] is the first development under the quadrature rule-based filtering. It implements the product rule for extending the univariate Gauss-Hermite quadrature rule in multivariate domain. Subsequently, $I_0(f)$ is approximated as [38], [48]

$$I_0(f) \approx \sum_{i_1=1}^{N_{s1}} \cdots \sum_{i_n=1}^{N_{s1}} f(\left[q_{i_1}, q_{i_2}, \ldots, q_{i_n}\right]^T)\omega_{i_1}\omega_{i_2}\ldots\omega_{i_n} \tag{46}$$

where $\left[q_{i_1}, q_{i_2}, \ldots, q_{i_n}\right]^T$ $\forall$ $i_j \in \{1,2,\ldots,N_{s1}-1\}$ ($j \in \{1,2,\ldots,n\}$) are univariate quadrature points, and $\omega_{i_1}\omega_{i_2}\ldots\omega_{i_n}$ are associated weights, and T represents matrix transpose. Note that $\{q_1, q_2, \ldots, q_{N_{s_1}}\}$ and $\{\omega_1, \omega_2, \ldots, \omega_{N_{s_1}}\}$ can be determined using either of the moment matching method or Golub's method.

The generalized integral, $I(f)$, can be approximated as

$$I(f) \approx \sum_{i_1=1}^{N_{s1}} \cdots \sum_{i_n=1}^{N_{s1}} f(\hat{\mathbf{x}} + \mathbf{\Sigma}\left[q_{i_1}, q_{i_2}, \ldots, q_{i_n}\right]^T)\omega_{i_1}\omega_{i_2}\ldots\omega_{i_n}. \tag{47}$$

GHF requires $(N_{s1})^n$ quadrature points, which means that the sample point requirement increases exponentially with increasing dimension. Subsequently, it suffers from the "*curse of dimensionality*" problem. However, the high estimation accuracy and fast convergence of GHF attract the practitioners despite of the high computational burden. It finds several applications in defense and space technologies [84], where the computational budget is significantly high. Moreover, the efficacy of computational devices is improving with growing technology, which brings a hope of even broader applicability in the future.

C. Sparse-Grid Gauss-Hermite Filter

The SGHF is an attempt to reduce the *curse of dimensionality* problem associated with GHF, so that, the quadrature filters can be implemented with relatively higher dimension. In this regard, it replaces the product rule by Smolyak rule [42], [85] in extending the univariate Gauss-Hermite quadrature rule in multivariate domain.

The Smolyak rule [42], [85] implements a linear combination of tensor products based on a predefined accuracy level L. The accuracy level L means that the approximation of $f(\mathbf{x}) = \int x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} d\mathbf{x}$, where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, is exact if $\sum_{j=1}^{n} i_j \leq 2L-1$ [42]. Using this rule, the integral $I_0(f)$ could be approximated as

$$
\begin{aligned}
I_0(f) \approx & \sum_{a=L-n}^{L-1} (-1)^{L-1-a} C_{n-1}^{L-1-a} \\
& \times \sum_{\Phi \in N_a^n} (I_{i_1} \otimes I_{i_2} \otimes \cdots \otimes I_{i_n}) f(\mathbf{x})
\end{aligned}
\tag{48}
$$

where C represents binomial coefficient, i.e., $C_k^n = n!/k!(n-k)!$, $\otimes$ stands for tensor product, I_{i_j} is univariate Gauss-Hermite quadrature rule [42] with accuracy level $i_j \in \Phi$ if $\Phi \triangleq \{i_1, \ldots, i_n\}$ is a set of univariate accuracy levels. The index set N_q^n is defined as

$$
\begin{aligned}
N_a^n &= \left\{\Phi : \sum_{j=1}^{n} i_j = n+a\right\} && \text{for} \quad a \geq 0 \\
&= \{\} && \text{for} \quad a < 0
\end{aligned}
$$

where a is constant with $L-n \leq a \leq L-1$, and $\{\}$ represents a null set.

Let us assume $\mathbf{X}_{i_j}$ is a set of univariate quadrature points corresponding to I_{i_j}, then (48) can be simplified as

$$
\begin{aligned}
I_0(f) \approx & \sum_{a=L-n}^{L-1} \sum_{\Phi \in N_a^n} \sum_{q_{i_1} \in \mathbf{X}_{i_1}} \cdots \sum_{q_{i_n} \in \mathbf{X}_{i_n}} f\left([q_{i_1}, \ldots, q_{i_n}]^T\right) \\
& \times \underbrace{(-1)^{L-1-a} C_{n-1}^{L-1-a} \left(\omega_{i_1} \ldots \omega_{i_n}\right)}_{weight}.
\end{aligned}
$$

Subsequently, $I(f)$ can be approximated as

$$
\begin{aligned}
I(f) \approx & \sum_{a=L-n}^{L-1} \sum_{\Phi \in N_a^n} \sum_{q_{i_1} \in \mathbf{X}_{i_1}} \cdots \sum_{q_{i_n} \in \mathbf{X}_{i_n}} f(\hat{\mathbf{x}} + \mathbf{\Sigma} \times [q_{i_1}, \ldots, q_{i_n}]^T) \\
& \times \underbrace{(-1)^{L-1-a} C_{n-1}^{L-1-a} \left(\omega_{i_1} \ldots \omega_{i_n}\right)}_{weight}.
\end{aligned}
$$

The Smolyak rule-based SGHF retains the estimation accuracy of GHF at a significantly reduced computational cost. Subsequently, the high estimation accuracy and fast convergence of quadrature rule-based filters can be celebrated over a wider horizon. However, the computational burden is still substantial for an ordinary processor if the dimension of the system is beyond 5 or 6. Furthermore, the mathematical background is not as simple as the product rule-based GHF to understand.

D. Multiple Gauss-Hermite Filter

The MGHF [45] is another attempt to reduce the computational burden of GHF. Note that the computational burden of GHF is acceptable for lower dimensional systems, but it is arbitrarily large for higher dimensional systems (due to the curse of dimensionality problem). Therefore, the MGHF aims to preclude the high dimensionality by implementing the principle of state partitioning. Subsequently, it partitions the high dimensional state space in multiple subspaces with smaller dimensions. During the filtering, the subspaces are propagated independently by parallel processing of ordinary GHF.

Let us assume $\mathbf{x}_k^{(1)}$, $\mathbf{x}_k^{(2)}$, …, $\mathbf{x}_k^{(S)}$ are independent subspaces of $\mathbf{x}_k$, then the process model could be partitioned as

$$
\mathbf{x}_k = \begin{pmatrix} \mathbf{x}_k^{(1)} \\ \mathbf{x}_k^{(2)} \\ \vdots \\ \mathbf{x}_k^{(S)} \end{pmatrix} = \begin{pmatrix} \phi_{k-1}^{(1)}(\mathbf{x}_{k-1}^{(1)}) \\ \phi_{k-1}^{(2)}(\mathbf{x}_{k-1}^{(2)}) \\ \vdots \\ \phi_{k-1}^{(S)}(\mathbf{x}_{k-1}^{(S)}) \end{pmatrix} + \begin{pmatrix} \boldsymbol{\eta}_k^{(1)} \\ \boldsymbol{\eta}_k^{(2)} \\ \vdots \\ \boldsymbol{\eta}_k^{(S)} \end{pmatrix}
\tag{49}
$$

where $\phi_{k-1}^{(s)} : \mathbf{x}_{k-1}^s \to \mathbf{x}_k^s \quad \forall \quad s \in \{1, 2, \ldots, S\}$ are nonlinear functions representing the state dynamics for sth subspace and $\boldsymbol{\eta}_k^{(s)} \sim \aleph(0, \mathbf{Q}^{(s)})$ is the process noise associated with sth subspace. $\mathbf{Q}^{(s)}$ is obtained from $\mathbf{Q}$ as the entries associated with sth subspace. If $n^{(s)}$ is the dimension of sth subspace, then $n^{(s)} < n \ \forall \ s \in \{1, 2, \ldots, S\}$ and $\sum_{s=1}^{S} n^{(s)} = n$.

The MGHF estimates $\mathbf{x}_k^{(1)}$, $\mathbf{x}_k^{(2)}$, …, $\mathbf{x}_k^{(S)}$ independently by parallel processing of ordinary GHF with dimensions $n^{(1)}$, $n^{(2)}$, …, $n^{(S)}$, respectively. As $n^{(s)} < n \ \forall \ s \in \{1, 2, \ldots, S\}$, the partitioning helps in reducing the high dimensional filtering problems into several lower dimensional filtering problems, which further helps in reducing the computational burden. In a similar development, the multiple SGHF (MSGHF) [46] is introduced by replacing GHF with SGHF to reduce the computational burden further.

The state partitioning assumes the subspaces to be independent of each other, though they are dependent and correlated in most of the practical problems. Subsequently, the MGHF and MSGHF may not be applicable to many of the real-life filtering problems.

E. Adaptive Sparse-Grid Gauss-Hermite Filter

The ASGHF [47] introduced an adaptive approach for generating a reduced number of quadrature points depending on the degree of nonlinearity along different dimensions. The adaptive approach generates an n-dimensional admissible index set $\mathbb{I}_n$, with a reduced number of indices compared to N_a^n generated by the SGHF. The admissibility of $\mathbb{I}_n$ is defined as $\epsilon - e_j \in \mathbb{I}_n$; $\forall \ \epsilon \in \{\mathbb{I}_n - I_1\}$ with e_j being jth column of n-dimensional identity matrix and $I_1 = [1, 1, \ldots, 1]^T$ being the first entry in $\mathbb{I}_n$. Based on $\mathbb{I}_n$, the integral of interest $I_0(f)$ is approximated as [47], [86]

$$I_0(f) \approx \sum_{\epsilon\in I_n} \triangle_{\epsilon} f(\mathbf{x}) = \sum_{\epsilon\in I_n}(\triangle_{\epsilon_1}\otimes\cdots\otimes\triangle_{\epsilon_n})f(\mathbf{x}) \tag{50}$$

where $\triangle$ represents difference formula which is defined for a univariate function f^1 as $\triangle_i f^1 = (I_i - I_{i-1})f^1$ with $I_0 f^1 = 0$, and $\triangle_\epsilon$ stands for tensor product of $\triangle_{\epsilon_1}$, $\triangle_{\epsilon_2}$, …, $\triangle_{\epsilon_n}$ with $\epsilon = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_n\}$. The Smolyak rule, used by SGHF, is also a difference formula-based numerical approximation technique, but it is implemented over a relatively larger index set N_a^n compared to $\mathbb{I}_n$.

The construction of $\mathbb{I}_n$ begins with I_1, and addition of new points are based on local error indicator [47], [86]

$$g_\epsilon = \max\left\{\psi\frac{|\triangle_\epsilon f|}{|\triangle_{I_1} f|}, (1-\psi)\frac{\varpi_{I_1}}{\varpi_\epsilon}\right\} \tag{51}$$

where $|\triangle_\epsilon f|$ represents the first norm of absolute of $\triangle_\epsilon f$, $\psi \in [0,1]$ is error weighting parameter and ϖ_ϵ is the number of function evaluations for ϵ. The terms $\frac{|\triangle_\epsilon f|}{|\triangle_{I_1} f|}$ and $\frac{\varpi_{I_1}}{\varpi_\epsilon}$ are proxies to accuracy and computational load, respectively, while ψ bags a condition for achieving a trade-off between the accuracy and computational load. During the scanning of new indices, the adaptive approach examines the forward indices for their possible inclusion in $\mathbb{I}_n(\{[2,1,\ldots,1]^T,\ldots,[1,1,\ldots,2]^T\}$ are forward indices of I_1). A new index is first taken in an active index set A. After the inclusion possibility is examined, it is shifted to an old index set O. The index with smallest g in A is shifted to O if it satisfies the admissibility with latest O. A global error parameter is monitored, and the process of adding new indices is stopped after this parameter is smaller than a predefined tolerance level TOL.

The ASGHF is computationally most efficient among the existing quadrature rule-based filters. It reduces the computational burden without compromising much on the estimation accuracy. It also allows a fine-tuning of the number of sample points by varying ψ and TOL, which is helpful in close to optimal utilization of available computational budget. However, ψ and TOL vary with the system, and no specific rule is provided for their selection. Subsequently, sufficient offline analysis is required before going for online implementation to generate model specific ψ and TOL. Moreover, if the online model is mismatched with the offline training model, it may suffer from poor estimation accuracy.

VI. Filtering With Transformed Sample Points

Chang *et al.* [27] enhanced the estimation accuracy of UKF and CKF by orthogonal transformation of sample points. It was shown that a particular class of orthogonal transformation reduces the higher-order moments of Taylor series expansion around the estimates (see Appendix A for a better understanding of higher-order moments). The sum of higher-order moments is the residue, which quantifies the numerical approximation error. Thus, the residue reduction (by orthogonal transformation) leads to a better numerical approximation, which further leads to an enhanced estimation accuracy.

The initial objective is to derive an orthogonal transformation which maintains the orthogonality for varying system dimension (so that the transformation can be applied to systems with arbitrary dimension), and reduces the residue for changing system dynamics. Chang *et al.* introduced an orthogonal matrix $\mathbf{O}$ to serve the purpose and demonstrated its efficacy with a series of mathematical evidences [27]. If $\mathbf{O}_{i,j}$ is ith row and jth column of $\mathbf{O}$, then [27], [34]

1) $\mathbf{O}_{i,2j-1} = \sqrt{\frac{2}{n}}\cos((2j-1)i\pi/n)$; $\forall$ $i \in \{1,2,\ldots,n\}$ with $j \in \{1,2,\ldots,[n/2]\}$, where $[n/2]$ is the greatest integer not exceeding $n/2$.

2) $\mathbf{O}_{i,2j} = \sqrt{\frac{2}{n}}\sin((2j-1)i\pi/n)$.

3) If n is odd, $\mathbf{O}_{i,n} = (-1)^i/\sqrt{n}$.

Let us assume, $\boldsymbol{\xi}$ is the set of sample points for an ordinary nonlinear filter. Then the set of transformed sample points is $\mathbf{O}\times\boldsymbol{\xi}$. The weights associated with the sample points remain unchanged after the transformation.

The analysis and mathematical evidence presented by Chang *et al.* [27] were limited to the UKF and CKF only. Later, it was extended for the CQKF in [34], [49], with additional supporting mathematical evidence. A preliminary analysis of the transformation over GHF is provided in [48].

The transformed filtering is an interesting development as it enhances the estimation accuracy without adding extra computational budget. However, the mathematical evidences introduced until now are only applicable to the UKF, CKF, and CQKF. In future developments, it may be interesting to see if the transformation may be extended for other filters, like the quadrature rule-based filters.

VII. Square-Root Filtering Algorithm

The Gaussian filtering is constrained with a positive-definite requirement of error covariance matrices. Therefore, it fails in many real-life problems, where the positive-definiteness is not guaranteed. To counter this constraint and extend the filter applicability further, a square-root extension [22], [33], [39], [50]–[52] of Gaussian filtering is introduced in the literature, which is based on QR decomposition. The steps followed in implementing the square-root filtering are provided herewith.

A. Time Update

The time update step in square-root filtering computes $\hat{\mathbf{x}}_{k|k-1}$ and $\boldsymbol{\Sigma}_{k|k-1}$, instead of $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$. The steps involved in this computation are as followed [33]:

1) Compute $\hat{\mathbf{x}}_{k|k-1}$, similar to (9).

2) Construct a diagonal matrix $\mathbf{W}_{sr}$ with ith diagonal element being $\sqrt{\mathbf{W}_i}$.

3) Transform the ith sample point with posterior estimate and covariance

$$\hat{\mathbf{x}}_{k-1|k-1} + \boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{\xi}_i.$$

4) Propagate the transformed sample points through process model

$$\chi^i_{k|k-1} = \phi_{k-1}(\hat{\mathbf{x}}_{k-1|k-1} + \boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{\xi}_i).$$

5) Obtain the weighted error matrix

$$\chi^*_{k|k-1} = [\chi^1_{k|k-1} - \hat{\mathbf{x}}_{k-1|k-1}\ \ \chi^2_{k|k-1} - \hat{\mathbf{x}}_{k-1|k-1}\\ \cdots\ \ \chi^{N_s}_{k|k-1} - \hat{\mathbf{x}}_{k-1|k-1}]\mathbf{W}_{sr}.$$

6) Perform QR decomposition of $[\chi^*_{k|k-1}\quad \sqrt{\mathbf{Q}}]$, i.e.,

$$\bar{\mathbf{R}}_{\mathbf{x}} = qr\left([\mathcal{X}^*_{k|k-1} \quad \sqrt{\mathbf{Q}}]\right)$$

where $\bar{\mathbf{R}}_{\mathbf{x}}$ is upper triangular component of QR decomposition.

7) Predict the square-root of error covariance

$$\mathbf{\Sigma}_{k|k-1} = \bar{\mathbf{R}}_{\mathbf{x}}.$$

B. Measurement Update

The measurement update parameters are computed through the following steps [33]:

1) Transform the ith sample point with predicted estimate and covariance

$$\hat{\mathbf{x}}_{k|k-1} + \mathbf{\Sigma}_{k|k-1}\boldsymbol{\xi}_i.$$

2) Propagate the transformed sample points with measurement model

$$\bar{\mathbf{Y}}^i_{k|k-1} = \gamma_k(\hat{\mathbf{x}}_{k|k-1} + \mathbf{\Sigma}_{k|k-1}\boldsymbol{\xi}_i).$$

3) Compute $\hat{\mathbf{y}}_{k|k-1}$, similar to (11).

4) Compute weighted error matrix

$$\begin{aligned}\bar{\mathbf{Y}}^*_{k|k-1} = [&\bar{\mathbf{Y}}^1_{k|k-1} - \hat{\mathbf{y}}_{k|k-1} \ \ \bar{\mathbf{Y}}^2_{k|k-1} - \hat{\mathbf{y}}_{k|k-1} \\ &\dots \ \bar{\mathbf{Y}}^{N_s}_{k|k-1} - \hat{\mathbf{y}}_{k|k-1}]\mathbf{W}_{sr}.\end{aligned}$$

5) Perform QR decomposition of $[\bar{\mathbf{Y}}^*_{k|k-1} \quad \sqrt{R}]$, i.e.,

$$\bar{\mathbf{R}}_{\mathbf{y}} = qr\left([\bar{\mathbf{Y}}^*_{k|k-1} \quad \sqrt{\mathbf{R}}]\right)$$

with $\bar{\mathbf{R}}_{\mathbf{y}}$ being upper triangular component of the QR decomposition.

6) Compute the square-root of innovation covariance

$$\mathbf{\Sigma}_{\mathbf{y}_{k|k-1}} = \bar{\mathbf{R}}_{\mathbf{y}}.$$

7) Compute the cross-covariance matrix

$$\mathbf{P}_{\mathbf{xy}} = \mathcal{X}^*_{k|k-1}\bar{\mathbf{Y}}^*_{k|k-1}.$$

8) Compute the Kalman gain

$$\mathbf{K}_k = \left(\mathbf{P}_{\mathbf{xy}}/\mathbf{\Sigma}^T_{\mathbf{y}_{k|k-1}}\right)/\mathbf{\Sigma}_{\mathbf{y}_{k|k-1}}.$$

9) Compute the posterior estimate

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}). \tag{52}$$

10) Perform the following QR decomposition:

$$\bar{\mathbf{R}} = qr\left([\mathcal{X}^*_{k|k-1} - \mathbf{K}_k\bar{\mathbf{Y}}^*_{k|k-1} \quad \sqrt{\mathbf{Q}} \quad \mathbf{K}_k\sqrt{\mathbf{R}}]\right)$$

where $\bar{\mathbf{R}}$ is upper triangular component of the QR decomposition.

11) Obtain the square-root of posterior estimate

$$\mathbf{\Sigma}_{k|k} = \bar{\mathbf{R}}. \tag{53}$$

The objective of square-root filtering is limited to improving the filter applicability. It does not impact the estimation accuracy significantly. Moreover, the computational efficiency is also similar to that of the ordinary filtering algorithm. Despite these facts, it is highly useful as the practical problems often fail to satisfy the positive-definiteness criteria for the covariance matrices.

VIII. Gaussian-Sum Filtering

Gaussian-sum filtering [53], [54] approximates the unknown pdf by multiple Gaussian components, instead of a single Gaussian component used in ordinary Gaussian filters. Each Gaussian component is assigned with an individual weight. It is shown in [53] and [54] that the multiple Gaussian components approximate the unknown pdf with better precision, resulting in an improved estimation accuracy.

Let us assume $P(\mathbf{x}_{k-1}|\mathbf{y}_{k-1})$ is approximated with N_G Gaussian components, then [53], [54]

$$P(\mathbf{x}_{k-1}|\mathbf{y}_{k-1}) \approx \sum_{g=0}^{N_G-1} w^g_{k-1}\aleph(\mathbf{x}_{k-1};\hat{\mathbf{x}}^g_{k-1|k-1},\mathbf{P}^g_{k-1|k-1}) \tag{54}$$

where $\hat{\mathbf{x}}^g_{k-1|k-1}$ and $\mathbf{P}^g_{k-1|k-1}$ are estimate and covariance of gth Gaussian component, and w^g_{k-1} is the weight associated with gth component. The weights are non-negative and normalized, i.e., $w^g_{k-1} \geq 0$ and $\sum_{g=0}^{N_G-1} w^g_{k-1} = 1$.

The Gaussian-sum filtering consists of two steps: time update and measurement update.

A. Time Update

Let us recall (3)

$$P(\mathbf{x}_{k|k-1}) = \int P(\mathbf{x}_k|\mathbf{x}_{k-1})P(\mathbf{x}_{k-1}|\mathbf{y}_{k-1})d\mathbf{x}_{k-1}. \tag{55}$$

Substituting $P(\mathbf{x}_{k-1}|\mathbf{y}_{k-1})$ from (54), we get

$$\begin{aligned}p(\mathbf{x}_{k|k-1}) = \sum_{g=0}^{N_G-1} w^g_{k-1}\int P(\mathbf{x}_k|\mathbf{x}_{k-1})\aleph(\mathbf{x}_{k-1};\\ \hat{\mathbf{x}}^g_{k-1|k-1},\mathbf{P}^g_{k-1|k-1})d\mathbf{x}_{k-1}.\end{aligned} \tag{56}$$

Note that $P(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the likelihood of state $\mathbf{x}_k$ given by (1).

B. Measurement Update

Let us recall (4)

$$P(\mathbf{x}_{k|k}) = \frac{P(\mathbf{y}_k|\mathbf{x}_k)P(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{\int P(\mathbf{y}_k|\mathbf{x}_k)P(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k}. \tag{57}$$

It can be represented as [53], [54]

$$P(\mathbf{x}_{k|k}) \approx \sum_{g=0}^{N_G-1} w^g_k\aleph(\mathbf{x}_k;\hat{\mathbf{x}}^g_{k|k},\mathbf{P}^g_{k|k}) \tag{58}$$

where

$$w^g_k = \frac{P(\mathbf{y}_k|\mathbf{x}_k,g)w^g_{k-1}}{\sum_{g=0}^{N_G-1}P(\mathbf{y}_k|\mathbf{x}_k,g)w^g_{k-1}}. \tag{59}$$

The readers may refer to [53], [54] for a detailed discussion.

Any of the filters discussed earlier can be extended under the Gaussian-sum filtering approach to improve the estimation accuracy. It is advantageous in dealing with highly nonlinear systems, where the Gaussian assumed pdf takes an arbitrary shape after the transformation by nonlinear system models. The multiple Gaussian approximates this arbitrary shape pdf with better precision compared to the ordinary filters. The literature in recent years has witnessed many developments under the Gaussian-sum filtering [30], [54], [55] with motivation to improve the estimation accuracy and extend the filter applicability in adverse conditions.

IX. Continuous-Discrete Filtering

The traditional Bayesian framework is designed with discrete-time formulation of process dynamics, though the process dynamics generally follows a continuous-time physical law (e.g., in target tracking problems, the target motion follows the continuous laws of motion, but formulated in discrete time-domain). Note that the measurements are usually generated from a discrete source, hence a continuous-discrete formulation of systems dynamics (continuous process and discrete measurement) is more precise compared to a discrete-time formulation ((1) and (2)).

The discrete-time approximation of continuous process dynamics stands as a source of error, especially if the sampling interval is large, resulting in a reduced estimation accuracy. The filtering problems with large sampling intervals commonly appears in target tracking [3], navigation [87], stochastic control [88], etc.

The literature witnesses few developments [58], [59] to reduce the error encountered due to discrete-time formulation of continuous process dynamics. The class of such developments is popular as continuous-discrete filtering. In this filtering technique, the process is discretized at an interval much smaller than the sampling interval. A general approach used for the discretization is the Itô-Taylor expansion [58], [59], though the Fokker-Plank-Kolmogorov equation (FPKE) has also been used in recent developments [55], [89], [90]. The discussion in this review is limited to the Itô-Taylor expansion.

The literature on continuous-discrete filtering reports only a few noticeable developments [56], [58], [59], which reformulate the ordinary EKF, UKF, and CKF to perform with continuous-discrete formulation of state space model.

A. Continuous-Discrete State-Space Model

The continuous process model is mathematically formulated with the differential equation [59]

$$d\mathbf{x}(t) = f(\mathbf{x}(t), t)dt + \sqrt{\mathbf{Q}}dB(t) \tag{60}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is an n-dimensional state variable at time t, $f : \mathbb{R}^n \to \mathbb{R}^n$ is drift function, $\mathbf{Q}$ is diffusion matrix, and $B(t)$ is an n-dimensional standard Wiener process with increment $dB(t)$. The discrete-time measurement model remains the same as (2).

B. Discretization of Process Model

As discussed above, the process model is discretized at an interval much smaller than the sampling interval. The early literature [56], [58] used Itô-Taylor expansion of order 0.5 for the discretization, which is more commonly known as Euler's method. This method was later replaced by the Itô-Taylor expansion of order 1.5 [59], [91] to ensure a better approximation accuracy. The discussion in this review is limited to the Itô-Taylor expansion of order 1.5.

Using the Itô-Taylor expansion of order 1.5, the continuous process model is discretized as [59], [91]

$$\begin{aligned}\mathbf{x}(t+\delta) = \mathbf{x}(t) + \delta f(\mathbf{x}(t), t) + \frac{1}{2}\delta^2(\mathbb{L}_0 f(\mathbf{x}(t), t)) \\ + \sqrt{\mathbf{Q}}\overline{\mathbf{w}} + (\mathbb{L} f(\mathbf{x}(t), t))\overline{\mathbf{q}}\end{aligned} \tag{61}$$

where δ is the discretization interval, $\overline{\mathbf{w}} \in \mathbb{R}^n$ and $\overline{\mathbf{q}} \in \mathbb{R}^n$ are Gaussian random variables, and, $\mathbb{L}_0$ and $\mathbb{L}$ are given as

$$\mathbb{L}_0 = \frac{\partial}{\partial t} + \sum_{i=1}^{n} f_i \frac{\partial}{\partial \mathrm{x}_i} + \frac{1}{2}\sum_{j=1}^{n}\sum_{p=1}^{n}\sum_{q=1}^{n} \sqrt{\mathbf{Q}}_{p,j}\sqrt{\mathbf{Q}}_{q,j}\frac{\partial^2}{\partial \mathrm{x}_p \partial \mathrm{x}_q}$$

$$\mathbb{L} = \sum_{i=1}^{n}\sum_{j=1}^{n} \sqrt{\mathbf{Q}}_{i,j}\frac{\partial}{\partial \mathrm{x}_i}.$$

δ is a practitioner's choice, but it must ensure $T = m\delta$, i.e., $m \in \mathbb{Z}^+$ with $\mathbb{Z}^+$ being the set of natural numbers. $\overline{\mathbf{w}}$ and $\overline{\mathbf{q}}$ are zero mean Gaussian, i.e., $E[\overline{\mathbf{w}}\overline{\mathbf{w}}^T] = \delta\mathbf{I}_n$, $E[\overline{\mathbf{w}}\overline{\mathbf{q}}^T] = \frac{1}{2}\delta^2\mathbf{I}_n$, and $E[\overline{\mathbf{q}}\overline{\mathbf{q}}^T] = \frac{1}{3}\delta^3\mathbf{I}_n$.

A simplified form of (61) is [59]

$$\mathbf{x}(t+\delta) = f_d(\mathbf{x}(t), t) + \sqrt{\mathbf{Q}}\overline{\mathbf{w}} + (\mathbb{L} f(\mathbf{x}(t), t))\overline{\mathbf{q}} \tag{62}$$

where

$$f_d(\mathbf{x}(t), t) = \mathbf{x}(t) + \delta f(\mathbf{x}(t), t) + \frac{1}{2}\delta^2(\mathbb{L}_0 f(\mathbf{x}(t), t)) \tag{63}$$

is noise-free process model.

C. Continuous-Discrete Filtering

Similar to discrete-time filtering, the continuous-discrete filtering process consists of two steps [56], [58], [59]: time update and measurement update.

1) Time Update

The time update parameters at t_{k+1}, $\hat{\mathbf{x}}_{k+1|k}$, and $\mathbf{P}_{k+1|k}$, are obtained by performing m-step iteration of length δ (i.e., $m\delta = T$) between t_k and t_{k+1}. In this regard, m intermediate time-steps are defined between t_k and t_{k+1}, as $t_k + j\delta$ with $j \in \{1, 2, \ldots, m\}$. The estimate and covariance at jth intermediate step are obtained as

$$\hat{\mathbf{x}}_{k|k}^{j} \approx \int f_d(\mathbf{x}_{k+(j-1)\delta})\aleph(\mathbf{x}_{k+(j-1)\delta}; \hat{\mathbf{x}}_{k|k}^{j-1}, \mathbf{P}_{k|k}^{j-1})d\mathbf{x}_{k+(j-1)\delta} \tag{64}$$

$$\begin{aligned}\mathbf{P}_{k|k}^{j} \approx \int f_d(\mathbf{x}_{k+(j-1)\delta}) f_d(\mathbf{x}_{k+(j-1)\delta})^T \aleph(\mathbf{x}_{k+(j-1)\delta}; \\ \hat{\mathbf{x}}_{k|k}^{j-1}, \mathbf{P}_{k|k}^{j-1})d\mathbf{x}_{k+(j-1)\delta} - (\hat{\mathbf{x}}_{k|k}^{j-1})(\hat{\mathbf{x}}_{k|k}^{j-1})^T.\end{aligned} \tag{65}$$

An m-step iterative implementation of (64) and (65) gives

$$\hat{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_{k|k}^{m} \quad \text{and} \quad \mathbf{P}_{k+1|k} = \mathbf{P}_{k|k}^{m}.$$

2) Measurement Update

The measurement update steps are similar to discrete-time filtering, as presented in Section II-A-2).

It should be noted that any of the earlier discussed filters can be extended under the continuous-discrete filtering method. In the literature, the performance of EKF, UKF, and CKF have been explicitly tested with this extension, see [56]–[59] for detailed discussion.

X. Simulation

This section presents simulation results to validate the performance of different Gaussian filters. For this purpose, two nonlinear filtering problems are simulated in Matlab environment over a personal computer with Intel Core i7, 1.99 GHz processor, 8 GB RAM, and 64-bit operating system. The first problem is used for validating the performance of discrete-time filters; however, the other problem is used for validating the continuous-discrete filters.

A. Problem 1: Maneuvering Target Tracking With Discrete-Time System Model

This problem is a discrete-time maneuvering target tracking problem [28], [47]. The target follows a coordinated turn model, given as [28], [47]

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \boldsymbol{\eta}_k \tag{66}$$

where $\mathbf{x} = [x\ \dot{x}\ y\ \dot{y}\ \omega]^T$ with x and y denote displacement along two-dimensional Cartesian coordinates, respectively, ω is angular turn rate, and

$$\mathbf{F}_k = \begin{bmatrix} 1 & \dfrac{\sin(\omega_k T)}{\omega_k} & 0 & -\dfrac{1-\cos(\omega_k T)}{\omega_k} & 0 \\ 0 & \cos(\omega_k T) & 0 & -\sin(\omega_k T) & 0 \\ 0 & \dfrac{1-\cos(\omega_k T)}{\omega_k} & 1 & \dfrac{\sin(\omega_k T)}{\omega_k} & 0 \\ 0 & \sin(\omega_k T) & 0 & \cos(\omega_k T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The measurement consists of radial displacement and bearing angle. Therefore, $\mathbf{y}_k$ is modeled as [28], [47]

$$\mathbf{y}_k = \left[\sqrt{x_k^2 + y_k^2} \quad \text{atan2}(y_k, x_k) \right]^T + \mathbf{v}_k$$

where atan2 is the four quadrant inverse tangent function. The noise components, $\boldsymbol{\eta}_k$ and $\mathbf{v}_k$, are considered to be Gaussian with zero mean and covariance $\mathbf{Q}$ and $\mathbf{R}$, respectively. $\mathbf{Q}$ depends on sampling interval $T = 0.5$ s, with the full expression given in [47]. $\mathbf{R}$ is taken as $\text{diag}([\sigma_r^2\ \sigma_t^2])$ with $\sigma_r = 10$ m and $\sigma_t = \sqrt{10} \times 10^{-3}$ rad. A simulated dataset of true state is generated by considering the initial state $\mathbf{x}_0 = [1000\ \text{m}\ 30\ \text{m/s}\ 1000\ \text{m}\ 0\ \text{m/s}\ \omega]^T$ with $\omega = 6\ °/\text{s}$. The initial estimate of state is considered as normally distributed with mean $\mathbf{x}_0$ and covariance $\mathbf{P}_{0|0} = \text{diag}([200\ \text{m}^2\ \ 20\ \text{m}^2/\text{s}^2\ \ 200\ \text{m}^2\ \ 20\ \text{m}^2/\text{s}^2\ \ 10^{-4}\ \text{rad}^2/\text{s}^2])$. The filters are implemented over 100 time-steps. True and estimated (using CKF and GHF) trajectories of *x*-*y* position are shown in Fig. 2. The figure concludes a successful tracking of the moving target using CKF and GHF. Moreover, the mean of absolute relative error obtained over 200 Monte-Carlo simulations are shown in Table I for various discrete-time filters. A comparative analysis of absolute relative error with CKF and GHF concludes a successful tracking for other filters as well.

The further performance analysis is based on *RMSE*. The *RMSE* for position at *k*th instant is obtained as

$$RMSE_{\text{pos},k} = \sqrt{\frac{1}{M_c} \sum_{i=1}^{M_c} \left(\left(\mathbf{x}_k^i(1) - \hat{\mathbf{x}}_k^i(1) \right)^2 + \left(\mathbf{x}_k^i(3) - \hat{\mathbf{x}}_k^i(3) \right)^2 \right)} \tag{67}$$

where $M_c = 100$ is number of Monte-Carlo simulations, $k = 1, 2, ..., N$, and $\hat{\mathbf{x}}_k^i(j)$ represents the *j*th element of $\hat{\mathbf{x}}_k$ at *i*th Monte-Carlo simulation. The *RMSE* for velocity is computed similarly with elements of $\hat{\mathbf{x}}_k$ corresponding to the velocity.

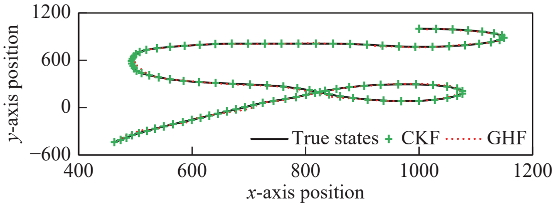


Fig. 2. Problem 1: True and estimated plots of *x*-*y* position.

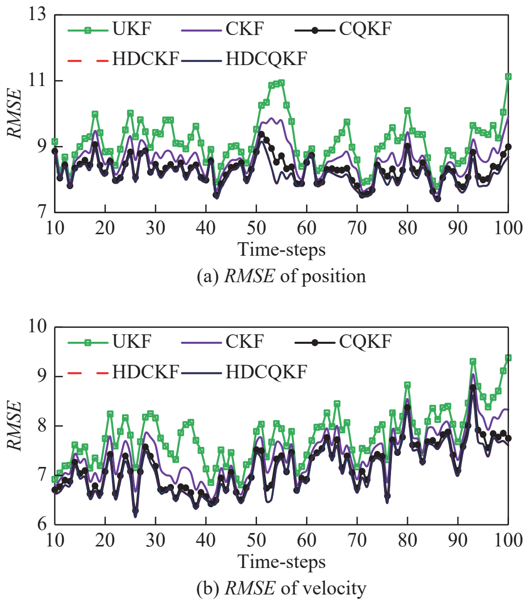
TABLE I
MEAN OF ABSOLUTE RELATIVE ERROR FOR VARIOUS DISCRETE-TIME FILTERS

Filter	State 1	State 2	State 3	State 4
UKF	0.00482666	0.22140210	0.00576063	0.22303688
CKF	0.00472118	0.21566884	0.00562553	0.21763431
CQKF	0.00464522	0.21176369	0.00552336	0.21344328
HDCKF	0.00459917	0.21011360	0.00548579	0.21145179
HDCQKF	0.00459996	0.21014015	0.00548637	0.21149179
GHF	0.00459999	0.21013906	0.00548621	0.21148573
SGHF	0.00459905	0.2101096	0.00548572	0.21144615
ASGHF	0.00459897	0.21010825	0.00548569	0.21144579
TCKF	0.00465111	0.2129956	0.00554156	0.21458457
TCQKF	0.00459798	0.21085377	0.00549281	0.21197251

The *RMSE* obtained from the cubature rule-based filters are compared with UKF in Fig. 3. The figure concludes a lower *RMSE*, i.e., an improved estimation accuracy, with cubature rule-based filters. Among the cubature rule-based filters, the *RMSE* decreases with successive developments, i.e., CQKF, HDCKF, and HDCQKF, which validates an improved estimation accuracy. However, the relative computational time is observed as 1, 1.03, 1.8, 3.29, and 4.98 for the UKF, CKF, CQKF, HDCKF, and HDCQKF, respectively. It indicates an increased computational burden for the successive development under cubature rule-based filtering.

The performance of quadrature rule-based filters is compared with the UKF and CKF in Fig. 4. The figure shows a reduced *RMSE* with quadrature rule-based filters, which validates an improved estimation accuracy. However, the relative computational time was observed as 1, 1.03, 13.26, 3.94, and 3.78 with UKF, CKF, GHF, SGHF, and ASGHF, respectively, which concludes an increased computational burden for the quadrature rule-based filters. Moreover, the analysis further concludes a successive reduction in computational time with SGHF and ASGHF (in comparison to GHF), while maintaining the estimation accuracy.

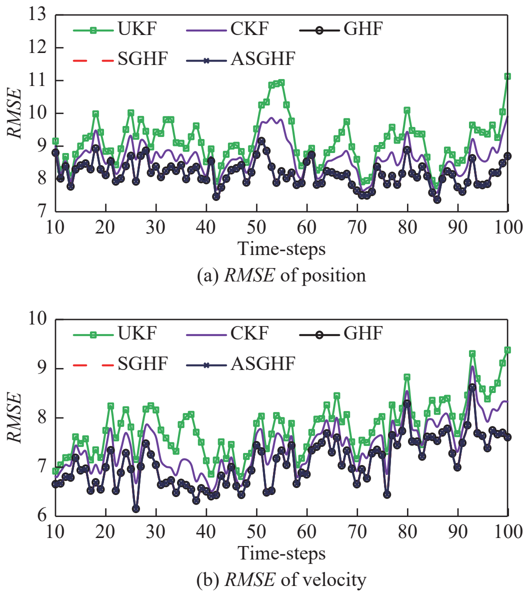
The performance of filtering with transformed sample points is analyzed in Fig. 5, which shows the RMSE of CKF and CQKF along with their transformed counterparts TCKF and TCQKF, respectively. The figure concludes a reduced RMSE with transformed filters, so that an improved estimation accuracy. Moreover, the computational time remains unchanged after the transformation. The analysis is limited to the CKF and CQKF only, as the literature witnesses a detailed study of transformation with these two filters only.



(a) *RMSE* of position

(b) *RMSE* of velocity

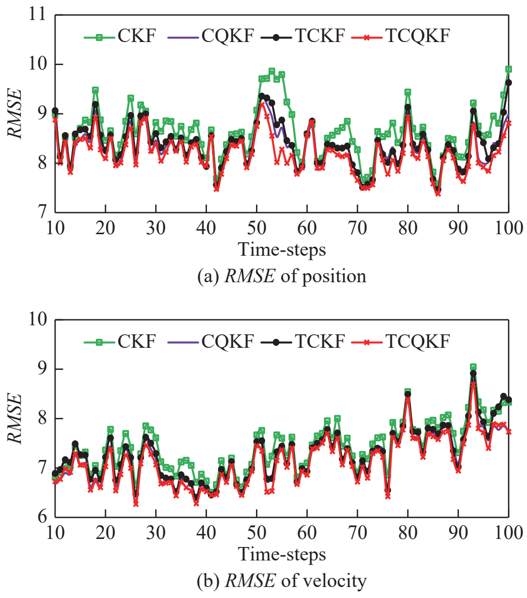
Fig. 3. Problem 1: *RMSE* plots for UKF and cubature rule-based filters.



(a) *RMSE* of position

(b) *RMSE* of velocity

Fig. 4. Problem 1: *RMSE* plots for UKF, CKF and quadrature rule-based filters.



(a) *RMSE* of position

(b) *RMSE* of velocity

Fig. 5. Problem 1: *RMSE* plots for ordinary and transformed CKF and CQKF.

B. Problem 2: Maneuvering Target Tracking With Continuous-Discrete System Model

This problem is simulated to analyze the performance of continuous-discrete filters. The problem considered here is an air-traffic control problem, where the height of the target is considered as constant. The continuous state dynamics of the target is modeled as [59]

$$d\mathbf{x}(t) = f(\mathbf{x}(t))dt + \sqrt{\mathbf{Q}}d\beta(t) \qquad (68)$$

where $\mathbf{x} = [\epsilon \;\; \dot\epsilon \;\; \eta \;\; \dot\eta \;\; \zeta \;\; \dot\zeta \;\; \omega]^T$ is state vector with ϵ, η, and ζ representing the displacement along the Cartesian coordinates x, y, and z, respectively, ω is turn rate, and the drift function $f(\mathbf{x})$ is [59]

$$f(\mathbf{x}) = [\dot\epsilon \;\; -\omega\dot\eta \;\; \dot\eta \;\; \omega\dot\epsilon \;\; \dot\zeta \;\; 0 \;\; 0]^T \qquad (69)$$

which shows a horizontal motion, $\beta(t) = [\beta_1(t) \; \beta_2(t) ... \; \beta_7(t)]^T$ is noise vector with $\beta_i(t)$ being standard Brownian motion independent of $\beta_j(t) \; \forall \; i \neq j$.

The continuous process model ((68)) is discretized at an interval much smaller than the sampling interval to characterize the continuity with high precision. The discretization is based on the Itô-Taylor expansion of order 1.5, which discretizes the process model as [59]

$$\mathbf{x}_k^{j+1} = f_d(\mathbf{x}_k^j) + \sqrt{\mathbf{Q}}\bar{\mathbf{w}} + (\mathbb{L}f(\mathbf{x}_k^j))\bar{\mathbf{q}} \qquad (70)$$

where $f_d(\mathbf{x})$ and $\mathbb{L}f(\mathbf{x})$ are nonlinear functions given in [59], and $\mathbf{Q} = \text{diag}([0 \;\; \sigma_1^2 \;\; 0 \;\; \sigma_1^2 \;\; 0 \;\; \sigma_1^2 \;\; \sigma_2^2)]$ with $\sigma_1 = \sqrt{2}\,\text{m}$ and $\sigma_2 = 2.85 \times 10^{-7} \; {}^{\text{o}}/\text{s}$.

The measurement equation is [59]

$$\begin{bmatrix} r_k \\ \theta_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} \sqrt{\epsilon_k^2 + \eta_k^2 + \zeta_k^2} \\ \tan^{-1}\left(\dfrac{\eta_k}{\epsilon_k}\right) \\ \tan^{-1}\left(\dfrac{\zeta_k}{\sqrt{\epsilon_k^2 + \eta_k^2}}\right) \end{bmatrix} + \mathbf{v}_k$$

where r is range, θ is elevation angle, ϕ is azimuth, and $\mathbf{v}_k \sim \aleph(0, \mathbf{R})$ is measurements noise; $\mathbf{R} = \text{diag}[(\sigma_r^2 \;\; \sigma_\theta^2 \;\; \sigma_\phi^2)]$ with $\sigma_r = 0.1\,\text{m}$, $\sigma_\theta = 0.1^{\text{o}}$, and $\sigma_\phi = 0.1^{\text{o}}$.

The true value of initial state is taken as $\mathbf{x}_0 = [1000 \; 0 \; 2650 \; 150 \; 200 \; 0 \; \omega]^T$ with $\omega = 1\,{}^\circ/\text{s}$, which indicates a constant height motion. The initial estimate is considered as normally distributed with mean $\mathbf{x}_0$ and covariance $\mathbf{P}_{0|0} = \text{diag}([100 \; 10 \; 100 \; 100 \; (\sqrt{0.1}\;\text{deg/s})^2])$. The simulation is performed over 250 s with sampling interval $T = 1\,\text{s}$ and the number of intermediate steps $m = 10$. The estimated trajectory of the x-y position obtained using the CD-UKF and CD-CKF are shown in Fig. 6, and are compared with the true trajectory. A close match of the true and estimated trajectories concludes a successful tracking using the CD-UKF and CD-CKF. The mean of absolute relative error for the CD-UKF and CD-CKF obtained over 200 Monte-Carlo executions are shown in Table II. The relative error is not analyzed along the height (state 5 and 6) as a constant height motion is considered. Moreover, the RMSEs obtained from the CD-UKF and CD-CKF are compared with their discrete-time counterparts, UKF and CKF, in Fig. 7. The figure concludes an arbitrarily large *RMSE*, such that divergence for the discrete-time filters, the UKF, and CKF.

The covariance matrices failed to satisfy the positive-definiteness criteria during the implementation of the continuous-discrete filters. Subsequently, the traditional filtering approaches failed. However, a square-root extension could help in successful implementation, which validates the importance of square-root filtering.

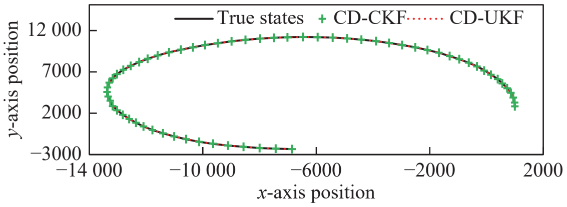
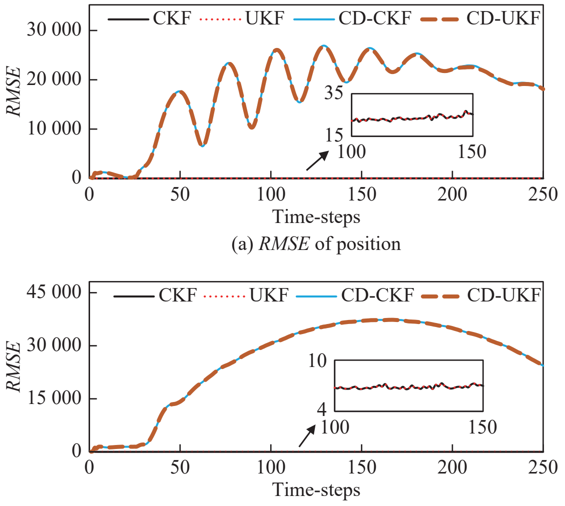


Fig. 6. Problem 2: True and estimated plots of x-y position for continuous-discrete filters.

TABLE II
MEAN OF ABSOLUTE RELATIVE ERROR FOR CONTINUOUS-DISCRETE FILTERS

Filter	State 1	State 2	State 3	State 4
CD-UKF	0.008912910	0.07939975	0.00829650	0.05744977
CD-CKF	0.008582399	0.07938377	0.00829650	0.05744935



(a) *RMSE* of position

(b) *RMSE* of velocity

Fig. 7. Problem 2: *RMSE* plots for CD-UKF, CD-CKF, and their discrete-time counterparts, UKF, and CKF.

XI. DISCUSSIONS AND CONCLUSIONS

The Bayesian framework of filtering has been a common choice among practitioners for several decades. It is a conceptual solution, and a commonly practiced analytical interpretation is Gaussian filtering, which offers high estimation accuracy at low computational cost. The early Gaussian filters, the EKF and its variants, suffered from several limitations due to derivative-based implementation. A derivative-free approach, named as UKF, was introduced in the nineties based on numerical approximation of intractable integrals. The UKF shows several advantages over the derivative-based filtering. Subsequently, the research continued further, leading to many derivative-free filters developed by advancing the numerical approximation techniques. This review classifies these developments in two categories: cubature rule-based filters and quadrature rule-based filters. The cubature rule-based filters decompose the intractable integrals into spherical and radial sub-parts. The spherical sub-part is approximated using the spherical-cubature rule, while the radial sub-part is approximated using the Gauss-Laguerre quadrature rule. Several developments appeared in this category by offering different order of approximation for the spherical and radial sub-parts. Some popular developments are CKF, CQKF, HDCKF, and HDCQKF. The quadrature rule-based filters, on the other side, implement a univariate Gauss-Hermite quadrature rule for the numerical approximation. An additional mathematical law is implemented for extending the univariate rule in multivariate domain. The filters in this category are known for high estimation accuracy, but often inapplicable due to high computational cost. The initial development was GHF, which is followed by several developments, like SGHF, ASGHF, and MGHF, to improve the computational efficiency.

The quadrature rule-based filters are preferred over the cubataure rule-based filters if the estimation accuracy is crucial and a large computational budget is allotted. On the other hand, the practitioners may choose to cut the computational budget by using cubature rule-based filters if the estimation accuracy is not very crucial. Both the cubature rule-based filtering and quadrature rule-based filtering offer different choices (with varying estimation accuracy and computational budget) to reach a trade-off between the desired estimation accuracy and the available computational budget.

The literature consists of some developments to modify the traditional Gaussian filtering approach and the Bayesian framework as well. Some popular modifications in the Gaussian filtering approach are square-root filtering and Gaussian-sum filtering. The square-root filtering aims to extend the filter applicability, while the Gaussian-sum filtering is motivated to improve the estimation accuracy. In the modification to the Bayesian framework, this paper reviews the continuous-discrete filtering methods.

APPENDIX A
ESTIMATES IN TERMS OF MOMENTS

The Gaussian filters implement a numerical approximation technique in the computation of estimates. The numerical approximation techniques are accurate only up to a particular order of Taylor series expansion. The higher-order terms contribute to the approximation error, which further contributes to estimation error. Subsequently, the representation of estimates in terms of moments helps in characterizing the error (the higher-order moments). Before proceeding forward, let us define the following notations:

1) For $\mathbf{a}$ being an n-dimensional vector and $l \in \{1,2,\ldots,\infty\}$

$$s(\mathbf{a},k,n,2l) = k\text{th term of } \left(\sum_{j=1}^{n}\mathbf{a}_j\right)^{2l} \tag{71}$$

where k is an integer.

2) $m_{s(\mathbf{a},k,n,2l)} = E[s(\mathbf{a},k,n,2l)]$ is the first moment of $s(\mathbf{a},k,n,2l)$.

3) $\sigma_{\mathbf{x}} = \boldsymbol{\xi}\sqrt{\mathbf{P}}$, where $\boldsymbol{\xi}$ is set of sample points defined for zero-mean and unity covariance system.

Let us expand $\mathbf{z} = f(\mathbf{x})$ around $\hat{\mathbf{x}}$ using Taylor series expansion

$$\mathbf{z} = f(\hat{\mathbf{x}}+\sigma_{\mathbf{x}}) = f(\hat{\mathbf{x}}) + (\sigma_{\mathbf{x}}^T\nabla)f(\hat{\mathbf{x}}) + \frac{1}{2!}(\sigma_{\mathbf{x}}^T\nabla)^2 f(\hat{\mathbf{x}}) + \cdots$$

where $\nabla = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n}\right]^T$. The expectation of $\mathbf{z}$ can be given as

$$\hat{\mathbf{z}} = f(\hat{\mathbf{x}}) + E[(\sigma_{\mathbf{x}}^T\nabla)]f(\hat{\mathbf{x}}) + \frac{1}{2!}E[(\sigma_{\mathbf{x}}^T\nabla)^2]f(\hat{\mathbf{x}}) + \cdots.$$

The symmetric property of Gaussian distribution nullifies the odd moments, hence,

$$\hat{\mathbf{z}} = f(\hat{\mathbf{x}}) + \sum_{l=1}^{\infty}\frac{1}{(2l)!}E[(\sigma_{\mathbf{x}}^T\nabla)^{(2l)}]f(\hat{\mathbf{x}}).$$

Substituting $\sigma_{\mathbf{x}}^T\nabla$, we get

$$\hat{\mathbf{z}} = f(\hat{\mathbf{x}}) + \sum_{l=1}^{\infty}\frac{1}{(2l)!}E\left[\left(\sum_{j=1}^{n}\sigma_{\mathbf{x}_{i,j}}\frac{\partial}{\partial x_j}\right)^{(2l)}\right]f(\hat{\mathbf{x}})$$

where

$$\begin{aligned}E\left[\left(\sum_{j=1}^{n}\sigma_{\mathbf{x}_{i,j}}\frac{\partial}{\partial x_j}\right)^{2l}\right] = E\Bigg[&\sigma_{\mathbf{x}_{i,1}}^{2l}\frac{\partial^{2l}}{\partial x_1^{2l}}\\ &+\sigma_{\mathbf{x}_{i,1}}^{2l-1}\sigma_{\mathbf{x}_{i,2}}\frac{\partial^{2l}}{\partial x_1^{2l-1}\partial x_2} + \sigma_{\mathbf{x}_{i,1}}^{2l-2}\sigma_{\mathbf{x}_{i,2}}^{2}\frac{\partial^{2l}}{\partial x_1^{2l-2}\partial x_2^2}\\ &+\sigma_{\mathbf{x}_{i,1}}^{2l-2}\sigma_{\mathbf{x}_{i,2}}\sigma_{\mathbf{x}_{i,3}}\frac{\partial^{2l}}{\partial x_1^{2l-2}\partial x_2\partial x_3} + \cdots\Bigg].\end{aligned}$$

Under the definition of $s(\mathbf{a},k,n,2l)$ and $m_{s(\mathbf{a},k,n,2l)}$, a simplified form of $\hat{\mathbf{z}}$ is obtained as

$$\hat{\mathbf{z}} = f(\hat{\mathbf{x}}) + \sum_{l=1}^{\infty}\frac{1}{(2l)!}\sum_{k=1}^{n^{2l}}\left(m_{s(\sigma_{\mathbf{x}},k,n,2l)}\frac{\partial^{2l}f(\hat{\mathbf{x}})}{s(\partial\mathbf{x},k,n,2l)}\right).$$

The moment terms for $l \geq 2$ represent higher-order moments.

Appendix B
Gaussian Sigma Point Kalman Filtering Algorithm

1) Prerequisites

i) The sample points and associated weights, $\boldsymbol{\xi}_j$ and $\mathbf{W}_j$ $\forall$ $j \in \{1,2,\ldots,N_s\}$ with N_s being the number of sample points.

ii) The initial estimate and covariance of state, $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$, respectively.

2) Step 1: Time Update

i) Obtain the square root of $\mathbf{P}_{k-1|k-1}$ by performing Cholesky decomposition

$$\mathbf{P}_{k-1|k-1} = \boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{\Sigma}_{k-1|k-1}^T.$$

ii) Transform the sample points by mean and covariance

$$\boldsymbol{\xi}_{j,k-1|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{\xi}_j.$$

iii) Propagate the transformed sample points through the process dynamics

$$\boldsymbol{\xi}_{j,k|k-1} = \phi_{k-1}(\boldsymbol{\xi}_{j,k-1|k-1}).$$

iv) Predict the estimate and error covariance for states

$$\hat{\mathbf{x}}_{k|k-1} = \sum_j \mathbf{W}_j\boldsymbol{\xi}_{j,k|k-1}$$

$$\begin{aligned}\mathbf{P}_{k|k-1} = \sum_j \mathbf{W}_j(&\boldsymbol{\xi}_{j,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\boldsymbol{\xi}_{j,k|k-1}\\ &- \hat{\mathbf{x}}_{k|k-1})^T + \mathbf{Q}.\end{aligned}$$

3) Step 2: Measurement Update

i) Obtain the square root of $\mathbf{P}_{k|k-1}$ by performing Cholesky decomposition

$$\mathbf{P}_{k|k-1} = \boldsymbol{\Sigma}_{k|k-1}\boldsymbol{\Sigma}_{k|k-1}^T.$$

ii) Transform the sample points with predicted mean and covariance

$$\boldsymbol{\xi}'_{j,k|k-1} = \hat{\mathbf{x}}_{k|k-1} + \boldsymbol{\Sigma}_{k|k-1}\boldsymbol{\xi}_j.$$

iii) Propagate the transformed sample points through the measurement equation

$$\bar{\mathbf{Y}}_{j,k|k-1} = \gamma_k(\boldsymbol{\xi}'_{j,k|k-1}).$$

iv) Obtain the predicted estimate and covariance of measurement

$$\hat{\mathbf{y}}_{k|k-1} = \sum_j \mathbf{W}_j\bar{\mathbf{Y}}_{j,k|k-1}$$

$$\begin{aligned}\mathbf{P}_{k|k-1}^{\mathbf{yy}} = \sum_j \mathbf{W}_j(&\bar{\mathbf{Y}}_{j,k|k-1} - \hat{\mathbf{y}}_{k|k-1})\\ &\times(\bar{\mathbf{Y}}_{j,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T + \mathbf{R}.\end{aligned}$$

v) Compute the cross-covariance between state and measurement

$$\mathbf{P}_{k|k-1}^{\mathbf{xy}} = \sum_j \mathbf{W}_j(\boldsymbol{\xi}_{j,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\bar{\mathbf{Y}}_{j,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T.$$

vi) Compute the Kalman gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}^{\mathbf{xy}}(\mathbf{P}_{k|k-1}^{\mathbf{yy}})^{-1}.$$

vii) Compute the posterior estimate and covariance for states

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}),$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{k|k-1}^{\mathbf{yy}}\mathbf{K}_k^T.$$

References

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With*

Applications to Tracking and Navigation. New York, USA: John Wiley & Sons, 2001.

[2] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. New York, USA: Wiley, 1997.

[3] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, USA: Prentice Hall, 1979.

[4] A. Gelb, *Applied Optimal Estimation*. Boston, USA: MIT Press, 1974.

[5] D. Simon, *Optimal State Estimation*. New Jersey, USA: Wiley, 2006.

[6] M. Abo, O. W. Marquez, J. McNames, R. Hornero, T. Trong, and B. Goldstein, "Adaptive modeling and spectral estimation of nonstationary biomedical signals based on Kalman filtering," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 8, pp. 1485–1489, Aug. 2005.

[7] C. Wells, "The Kalman filter in finance," *Springer*, vol. 32, 2013.

[8] P. L. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble kalman filter technique," *Mon. Wea. Rev.*, vol. 126, no. 3, pp. 796–811, Mar. 1998.

[9] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, USA: Cambridge University Press, 2013.

[10] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, USA: Dover Publications, 2007.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[12] D. Godard, "Channel equalization using a Kalman filter for fast data transmission," *IBM J. Res. Dev.*, vol. 18, no. 3, pp. 267–273, May 1974.

[13] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970.

[14] M. Athans, R. P. Wishner, and A. Bertolini, "Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements," *IEEE Trans. Autom. Control*, vol. 13, no. 5, pp. 504–514, Oct. 1968.

[15] S. K. Rao, "Modified gain extended Kalman filter with application to bearings-only passive manoeuvring target tracking," *IET Proc. Radar Sonar Navig.*, vol. 152, no. 4, pp. 239–244, Aug. 2005.

[16] T. Song and J. Speyer, "A stochastic analysis of a modified gain extended Kalman filter with applications to estimation with bearings only measurements," *IEEE Trans. Autom. Control*, vol. 30, no. 10, pp. 940–949, Oct. 1985.

[17] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*, 1997.

[18] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[19] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[20] S. J. Julier and J. K. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," in *Proc.American Control Conf.*, Anchorage, USA, 2002, pp. 887–892.

[21] S. J. Julier, "The scaled unscented transformation," in *Proc. American Control Conf.*, Anchorage, USA, 2002, pp. 4555–4559.

[22] R. van der Merwe and E. A. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, USA, 2001, pp. 3461–3464.

[23] R. van der Merwe and E. A. Wan, "Efficient derivative-free Kalman filters for online learning," in *Proc. European Symp. Artificial Neural Networks*, Bruges, Belgium, 2001, pp. 205–210.

[24] Y. X. Wu, D. W. Hu, M. P. Wu, and X. P. Hu, "Unscented Kalman filtering for additive noise case: Augmented versus nonaugmented," *IEEE Signal Process. Lett.*, vol. 12, no. 5, pp. 357–360, May 2005.

[25] Y. Lei, D. Xia, K. Erazo, and S. Nagarajaiah, "A novel unscented Kalman filter for recursive state-input-system identification of nonlinear systems," *Mech. Sys. Sign. Proc.*, vol. 127, pp. 120–135, Jul. 2019.

[26] Z. Jiang, Q. Song, Y. Q. He, and J. D. Han, "A novel adaptive unscented Kalman filter for nonlinear estimation," in *Proc. 46th IEEE Conf. Decision and Control*, New Orleans, USA, 2007, pp. 4293–4298.

[27] L. B. Chang, B. Q. Hu, A. Li, and F. J. Qin, "Transformed unscented Kalman filter," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 252–257, Jan. 2013.

[28] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.

[29] J. Mu and Y. L. Cai, "Iterated cubature Kalman filter and its application," in *Proc. IEEE Int. Conf. Cyber Tech. in Automation, Control, and Intelligent Systems*, Kunming, China, 2011, pp. 33–37.

[30] P. H. Leong, S. Arulampalam, T. H. Lamahewa, and T. D. Abhayapala, "A Gaussian-sum based cubature Kalman filter for bearings-only tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 1161–1176, Apr. 2013.

[31] S. Y. Wang, J. C. Feng, and C. K. Tse, "Spherical simplex-radial cubature Kalman filter," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 43–46, Jan. 2014.

[32] S. Bhaumik and Swati, "Cubature quadrature Kalman filter," *IET Signal Process.*, vol. 7, no. 7, pp. 533–541, Sept. 2013.

[33] S. Bhaumik and Swati, "Square-root cubature-quadrature Kalman filter," *Asian J. Control*, vol. 16, no. 2, pp. 617–622, Mar. 2014.

[34] A. K. Singh and S. Bhaumik, "Transformed cubature quadrature Kalman filter," *IET Sign. Process.*, vol. 11, no. 9, pp. 1095–1103, Jul. 2017.

[35] B. Jia, M. Xin, and Y. Cheng, "High-degree cubature Kalman filter," *Automatica*, vol. 49, no. 2, pp. 510–518, Feb. 2013.

[36] A. K. Singh and S. Bhaumik, "Higher degree cubature quadrature Kalman filter," *Int. J. Control Autom. Syst.*, vol. 13, no. 5, pp. 1097–1105, Oct. 2015.

[37] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 910–927, May 2000.

[38] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature," *Proc. IEEE*, vol. 95, no. 5, pp. 953–977, May 2007.

[39] I. Arasaratnam and S. Haykin, "Square-root quadrature Kalman filtering," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2589–2593, Jun. 2008.

[40] H. Singer, "Generalized gauss-hermite filtering," *AStA Adv. Stat. Anal.*, vol. 92, no. 2, pp. 179–195, May 2008.

[41] H. Singer, "Conditional Gauss-Hermite filtering with application to volatility estimation," *IEEE Trans. Autom. Contr.*, vol. 60, no. 9, pp. 2476–2481, Sept. 2015.

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

[43] B. Jia, M. Xin, and Y. Cheng, "Sparse Gauss-Hermite quadrature filter with application to spacecraft attitude estimation," *J. Guid. Control Dyn.*, vol. 34, no. 2, pp. 367–379, Mar-Apr. 2011.

[44] B. Jia, M. Xin, and Y. Cheng, "Comparison of the sparse-grid quadrature rule and the cubature rule in nonlinear filtering," in *Proc. IEEE 51st IEEE Conf. Decision and Control*, Maui, USA, 2012, pp. 6022–6027.

[45] P. Closas, C. Fernandez-Prades, and J. Vila-Valls, "Multiple quadrature Kalman filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6125–6137, Dec. 2012.

[46] R. Radhakrishnan, A. K. Singh, S. Bhaumik, and N. K. Tomar, "Multiple sparse-grid Gauss-Hermite filtering," *Appl. Math. Model.*, vol. 40, no. 7-8, pp. 4441–4450, Apr. 2016.

[47] A. K. Singh, R. Radhakrishnan, S. Bhaumik, and P. Date, "Adaptive sparse-grid Gauss–Hermite filter," *J. Comput. Appl. Math.*, vol. 342, pp. 305–316, Nov. 2018.

[48] A. K. Singh and S. Bhaumik, "Nonlinear estimation using transformed Gauss-Hermite quadrature points," in *Proc. IEEE Int. Conf. Signal Processing, Computing and Control*, Solan, India, 2013, pp. 1–4.

[49] A. K. Singh, S. Bhaumik, and R. Radhakrishnan, "Nonlinear estimation with transformed cubature quadrature points, " in *Proc. IEEE Int. Symp. Signal Processing and Information Technology*, Noida, India, 2014, pp. 428–432.

[50] D. Potnuru, K. P. B. Chandra, I. Arasaratnam, D. W. Gu, K. A. Mary, and S. B. Ch, "Derivative-free square-root cubature Kalman filter for non-linear brushless DC motors," *IET Electric Power Appl.*, vol. 10, no. 5, pp. 419–429, Apr. 2016.

[51] K. P. B. Chandra, D. W. Gu, and I. Postlethwaite, "Square root cubature information filter," *IEEE Sens. J.*, vol. 13, no. 2, pp. 750–758, Feb. 2013.

[52] X. J. Tang, Z. B. Liu, and J. S. Zhang, "Square-root quaternion cubature Kalman filtering for spacecraft attitude estimation," *Acta Astronaut.*, vol. 76, pp. 84–94, Jul-Aug. 2012.

[53] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Autom. Control*, vol. 17, no. 4, pp. 439–448, Aug. 1972.

[54] P. H. Leong, S. Arulampalam, T. A. Lamahewa, and T. D. Abhayapala, "Gaussian-sum cubature Kalman filter with improved robustness for bearings-only tracking," *IEEE Signal Process. Lett.*, vol. 21, no. 5, pp. 513–517, May 2014.

[55] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Adaptive Gaussian sum filter for nonlinear Bayesian estimation," *IEEE Trans. Autom. Control*, vol. 56, no. 9, pp. 2151–2156, Sept. 2011.

[56] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, USA: Academic, 1970.

[57] J. B. Jorgensen, P. G. Thomsen, H. Madsen, and M. R. Kristensen, "A computationally efficient and robust implementation of the continuous-discrete extended Kalman filter," in *Proc. American Control Conf.*, New York, USA, 2007, pp. 3706–3712.

[58] S. Sarkka, "On unscented Kalman filtering for state estimation of continuous-time nonlinear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1631–1641, Sept. 2007.

[59] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature Kalman filtering for continuous-discrete systems: theory and simulations," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 4977–4993, Oct. 2010.

[60] G. Y. Kulikov and M. V. Kulikova, "Accurate numerical implementation of the continuous-discrete extended Kalman filter," *IEEE Trans. Auto. Cont.*, vol. 59, no. 1, pp. 273–279, Jan. 2014.

[61] S. C. Patwardhan, S. Narasimhan, P. Jagadeesan, B. Gopaluni, and S. L. Shah, "Nonlinear Bayesian state estimation: A review of recent developments," *Control Eng. Pract.*, vol. 20, no. 10, pp. 933–953, Oct. 2012.

[62] H. Z. Fang, N. Tian, Y. B. Wang, M. C. Zhou, and M. A. Haile, "Nonlinear Bayesian estimation: From Kalman filtering to a broader horizon," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 2, pp. 401–417, Mar. 2018.

[63] H. Kushner, "Approximations to optimal nonlinear filters," *IEEE Trans. Autom. Control*, vol. 12, no. 5, pp. 546–556, Oct. 1967.

[64] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[65] R. van der Merwe and E. Wan, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," in *Proc. Workshop on Advances in Machine Learning*. 2004.

[66] C. S. Manohar and D. Roy, "Monte Carlo filters for identification of nonlinear structural dynamical systems," *Sadhana*, vol. 31, no. 4, pp. 399–427, Aug. 2006.

[67] V. K. Mishra, R. Radhakrishnan, A. K. Singh, and S. Bhaumik, "Bayesian filters for parameter identification of duffing oscillator," *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 425–430, Jan. 2018.

[68] R. H. Zhan and J. W. Wan, "Iterated unscented Kalman filter for passive target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 1155–1163, Jul. 2007.

[69] S. Jafarzadeh, C. Lascu, and M. S. Fadali, "State estimation of induction motor drives using the unscented Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4207–4216, Nov. 2012.

[70] P. H. Li, T. W. Zhang, and B. Ma, "Unscented Kalman filter for visual curve tracking," *Image Vision Comput.*, vol. 22, no. 2, pp. 157–164, Feb. 2004.

[71] M. Sepasi and F. Sassani, "On-line fault diagnosis of hydraulic systems using unscented Kalman filter," *Int. J. Control Autom. Syst.*, vol. 8, no. 1, pp. 149–156, Feb. 2010.

[72] A. Genz, "Fully symmetric interpolatory rules for multiple integrals over hyper-spherical surfaces," *J. Comput. Appl. Math.*, vol. 157, no. 1, pp. 187–195, Aug. 2003.

[73] A. H. Stroud, *Approximate Calculation of Multiple Integrals*. Englewood Cliffs, USA: Prentice-Hall, 1971.

[74] V. I. Krylov, *Approximate Calculation of Integrals*. New York, USA: Dover, 2006.

[75] F. B. Hildebrand, *Introduction to Numerical Analysis*. 2nd ed. New York, USA: McGraw Hill, 1973.

[76] J. Zarei, E. Shokri, and H. R. Karimi, "Convergence analysis of cubature Kalman filter," in *Proc. European Control Conf.*, Strasbourg, France, 2014.

[77] M. Havlicek, K. J. Friston, J. Jan, M. Brazdil, and V. D. Calhoun, "Dynamic modeling of neuronal responses in fMRI using cubature Kalman filtering," *NeuroImage*, vol. 56, no. 4, pp. 2109–2128, Jun. 2011.

[78] W. L. Li and Y. M. Jia, "Location of mobile station with maneuvers using an IMM-based cubature Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4338–4348, Nov. 2012.

[79] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on Gaussian filters," *IEEE Trans. Signal Process.*, vol. 54, no. 8, pp. 2910–2921, Aug. 2006.

[80] R. Cools and P. Rabinowitz, "Monomial cubature rules since "Stroud": A compilation," *J. Comput. Appl. Math.*, vol. 48, no. 3, pp. 309–326, Nov. 1993.

[81] F. B. Hildebrand, *Introduction to Numerical Analysis*. 2nd ed. St. Mineola, USA: Dover, 1987.

[82] S. S. Bonan and D. S. Clark, "Estimates of the Hermite and the Freud polynomials," *J. Approx. Theory*, vol. 63, no. 2, pp. 210–224, Nov. 1990.

[83] Y. H. Ku and M. Drubin, "Network synthesis using legendre and Hermite polynomials," *J. Frank. Inst.*, vol. 273, no. 2, pp. 138–157, Feb. 1962.

[84] A. K. Singh, K. Kumar, Swati, and S. Bhaumik "Cubature and quadrature based continuous-discrete filters for maneuvering target tracking," in *Proc. 21st Int. Conf. Information Fusion*, Cambridge, 2018.

[85] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Dokl. Akad. Nauk SSSR*, vol. 148, no. 5, pp. 1042–1045, 1963.

[86] T. Gerstner and M. Griebel, "Dimension-adaptive tensor-product quadrature," *Computing*, vol. 71, no. 1, pp. 65–87, Aug. 2003.

[87] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. Hoboken, USA: Wiley, 2001.

[88] K. J. Astrom, *Introduction to Stochastic Control Theory*. New York, USA: Dover, 1970.

[89] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models," *J. Guid. Control Dyn.*, vol. 31, no. 6, pp. 1623–1633, Nov.-Dec. 2008.

[90] M. Kumar, S. Chakravorty, P. Singla, and J. L. Junkins, "The partition of unity finite element approach with hp-refinement for the stationary Fokker-Planck equation," *J. Sound Vibr.*, vol. 327, no. 1-2, Oct. 2009.

[91] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Berlin, Germany: Springer, 1991.

Abhinoy Kumar Singh (M'19) received the B.Tech. degree in electrical and electronics engineering from Cochin University of Science and Technology (CUSAT), India, in 2012. He received the Ph.D. degree in electrical engineering from the Indian Institute of Technology Patna, India, in August 2016. During the Ph.D., he worked on algorithm development for nonlinear estimation and filtering. Post Ph.D., he initially worked at Shiv Nadar University, India, as an Assistant Prof. between December 2016 to April 2017. After that, he moved to Canada to pursue postdoctoral research at McGill University between May 2017 to May 2018. During the postdoctoral research, he worked on developing a continuous glucose monitoring (CGM) system, which is part of an underdeveloped artificial pancreas, with the application of Kalman filter. He is currently working at the Department of Electrical Engineering at the Indian Institute of Technology Indore, India, as an Inspire Faculty. His current research interests include estimation, filtering, and CGM systems.