Cubature Kalman Filters

Ienkaran Arasaratnam, and Simon Haykin, *Fellow, IEEE*

***Abstract*—In this paper, we present a new nonlinear filter for high-dimensional state estimation, which we have named the *cubature Kalman filter* (CKF). The heart of the CKF is a spherical-radial cubature rule, which makes it possible to numerically compute multivariate moment integrals encountered in the nonlinear Bayesian filter. Specifically, we derive a third-degree spherical-radial cubature rule that provides a set of cubature points scaling linearly with the state-vector dimension. The CKF may therefore provide a systematic solution for high-dimensional nonlinear filtering problems. The paper also includes the derivation of a square-root version of the CKF for improved numerical stability. The CKF is tested experimentally in two nonlinear state estimation problems. In the first problem, the proposed cubature rule is used to compute the second-order statistics of a nonlinearly transformed Gaussian random variable. The second problem addresses the use of the CKF for tracking a maneuvering aircraft. The results of both experiments demonstrate the improved performance of the CKF over conventional nonlinear filters.**

***Index Terms*—Bayesian filters, Cubature rules, Gaussian quadrature rules, Invariant theory, Kalman filter, Nonlinear filtering.**

I. Introduction

In this paper, we consider the filtering problem of a nonlinear dynamic system with additive noise, whose state-space model is defined by the pair of difference equations in discrete-time [1]:

$$\begin{aligned}\text{Process equation:} \quad \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{v}_{k-1} &(1)\\ \text{Measurement equation:} \quad \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, &(2)\end{aligned}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state of the dynamic system at discrete time k; $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ and $\mathbf{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_z}$ are some known functions; $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the known control input, which may be derived from a compensator as in Fig. 1; $\mathbf{z}_k \in \mathbb{R}^{n_z}$ is the measurement; $\{\mathbf{v}_{k-1}\}$ and $\{\mathbf{w}_k\}$ are independent process and measurement Gaussian noise sequences with zero means and covariances Q_{k-1} and R_k, respectively.

In the Bayesian filtering paradigm, the posterior density of the state provides a complete statistical description of the state at that time. On the receipt of a new measurement at time k, we update the old posterior density of the state at time $(k-1)$ in two basic steps:

- *Time update,* which involves computing the predictive density

$$\begin{aligned}p(\mathbf{x}_k|D_{k-1}) &= \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_k, \mathbf{x}_{k-1}|D_{k-1}) d\mathbf{x}_{k-1}\\ &= \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_{k-1}|D_{k-1})\\ &\quad \times p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) d\mathbf{x}_{k-1}, \qquad (3)\end{aligned}$$

where $D_{k-1} = \{\mathbf{u}_i, \mathbf{z}_i\}_{i=1}^{(k-1)}$ denotes the history of input-measurement pairs up to time $(k-1)$; $p(\mathbf{x}_{k-1}|D_{k-1})$ is the old posterior density at time $(k-1)$ and the state transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ is obtained from (1).

- *Measurement update,* which involves computing the posterior density of the current state:

$$p(\mathbf{x}_k|D_k) = p(\mathbf{x}_k|D_{k-1}, \mathbf{u}_k, \mathbf{z}_k).$$

Using the state-space model (1)-(2) and Bayes' rule we have

$$p(\mathbf{x}_k|D_k) = \frac{1}{c_k} p(\mathbf{x}_k|D_{k-1}, \mathbf{u}_k) p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k), \quad (4)$$

where the normalizing constant c_k is given by

$$\begin{aligned}c_k &= p(\mathbf{z}_k|D_{k-1}, \mathbf{u}_k)\\ &= \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_k|D_{k-1}, \mathbf{u}_k) p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k) d\mathbf{x}_k.\end{aligned}$$

To develop a recursive relationship between predictive and posterior densities in (4), the inputs have to satisfy the relationship

$$p(\mathbf{u}_k|D_{k-1}, \mathbf{x}_k) = p(\mathbf{u}_k|D_{k-1}),$$

which is also called the *natural condition of control* [2]. This condition therefore suggests that D_{k-1} has sufficient information to generate the input $\mathbf{u}_k$. To be specific, the input $\mathbf{u}_k$ can be generated using $\hat{\mathbf{x}}_{k|k-1}$. Under this condition, we may equivalently write

$$p(\mathbf{x}_k|D_{k-1}, \mathbf{u}_k) = p(\mathbf{x}_k|D_{k-1}). \quad (5)$$

Hence, substituting (5) into (4) yields

$$p(\mathbf{x}_k|D_k) = \frac{1}{c_k} p(\mathbf{x}_k|D_{k-1}) p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k), \quad (6)$$

as desired, where

$$c_k = \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_k|D_{k-1}) p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k) d\mathbf{x}_k, \quad (7)$$

and the measurement likelihood function $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k)$ is obtained from (2).

The Bayesian filter solution given by (3), and (6)-(7)

The authors are with the Cognitive Systems Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, L8S 4K1, Canada. Tel: 905 525 9140 X 27282, Fax: 905 521 2922, Email: aienkaran@grads.ece.mcmaster.ca, and haykin@mcmaster.ca.

The work was supported by the Natural Sciences & Engineering Research Council (NSERC) of Canada.

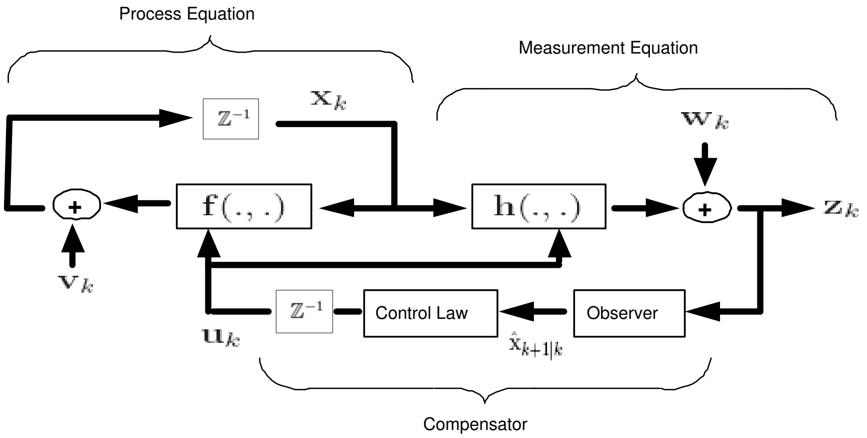


Fig. 1. Signal-flow diagram of a dynamic state-space model driven by the feedback control input. The observer may employ a Bayesian filter. The label $\mathbb{Z}^{-1}$ denotes the unit delay.

provides a unified recursive approach for nonlinear filtering problems, at least conceptually. From a practical perspective, however, we find that the multi-dimensional integrals involved in (3) and (7) are typically intractable. Notable exceptions arise in the following restricted cases:

1) A linear-Gaussian dynamic system, the optimal solution for which is given by the celebrated Kalman filter [3].
2) A discrete-valued state-space with a fixed number of states, the optimal solution for which is given by the grid filter (Hidden-Markov model filter) [4].
3) A ‘Benes type’ of nonlinearity, the optimal solution for which is also tractable [5].

In general, when we are confronted with a nonlinear filtering problem, we have to abandon the idea of seeking an optimal or analytical solution and be content with a suboptimal solution to the Bayesian filter [6]. In computational terms, suboptimal solutions to the posterior density can be obtained using one of two approximative approaches:

1) *Local approach.* Here, we derive nonlinear filters by fixing the posterior density to take *a priori* form. For example, we may assume it to be Gaussian; the nonlinear filters, namely, the extended Kalman filter (EKF) [7], the central-difference Kalman filter (CDKF) [8], [9], the unscented Kalman filter (UKF) [10], and the quadrature Kalman filter (QKF) [11], [12], fall under this first category. The emphasis on locality makes the design of the filter simple and fast to execute.
2) *Global approach.* Here, we do not make any explicit assumption about the posterior density form. For example, the point-mass filter using adaptive grids [13], the Gaussian mixture filter [14], and particle filters using Monte Carlo integrations with the importance sampling [15], [16] fall under this second category. Typically, the global methods suffer from enormous computational demands.

Unfortunately, the presently known nonlinear filters mentioned above suffer from the *curse of dimensionality* [17] or divergence or both. The effect of curse of dimensionality may often become detrimental in high-dimensional state-space models with state-vectors of size 20 or more. The divergence may occur for several reasons including (i) inaccurate or incomplete model of the underlying physical system, (ii) information loss in capturing the true evolving posterior density completely, e.g., a nonlinear filter designed under the Gaussian assumption may fail to capture key features of a multi-modal posterior density, (iii) high degree of nonlinearities in the equations that describe the state-space model, and (iv) numerical errors. Indeed, each of the above-mentioned filters has its own domain of applicability and it is doubtful that a single filter exists that would be considered effective for a complete range of applications. For example, the EKF, which has been the method of choice for nonlinear filtering problems in many practical applications for the last four decades, works well only in a ‘mild’ nonlinear environment owing to the first-order Taylor series approximation for nonlinear functions.

The motivation for this paper has been to derive a more accurate nonlinear filter that could be applied to solve a wide range (from low to high dimensions) of nonlinear filtering problems. Here, we take the local approach to build a new filter, which we have named the *cubature Kalman filter* (CKF). It is known that the Bayesian filter is rendered tractable when all conditional densities are assumed to be Gaussian. In this case, the Bayesian filter solution reduces to computing multi-dimensional integrals, whose integrands are all of the form *nonlinear function* $\times$ *Gaussian*. The CKF exploits the properties of highly efficient numerical integration methods known as cubature rules for those multi-dimensional integrals [18]. With the cubature rules at our disposal, we may describe the underlying philosophy behind the derivation of the new filter as *nonlinear filtering through linear estimation theory*, hence the name ‘cubature Kalman filter’. The CKF is numerically accurate and easily extendable to high-dimensional problems.

The rest of the paper is organized as follows: Section II derives the Bayesian filter theory in the Gaussian domain. Section III describes numerical methods available for moment integrals encountered in the Bayesian filter. The cubature Kalman filter, using a third-degree spherical-radial cubature rule, is derived in Section IV. Our argument for choosing a third-degree rule is articulated in Section V. We go on to derive a square-root version of the CKF for improved numerical stability in Section VI. The existing sigma-point approach is compared with the cubature method in Section VII. We apply the CKF in two nonlinear state estimation problems in Section VIII. Section IX concludes the paper with a possible extension of the CKF algorithm for a more general setting.

II. Bayesian Filter Theory in the Gaussian Domain

The key approximation taken to develop the Bayesian filter theory under the Gaussian domain is that the predictive density $p(\mathbf{x}_k|D_{k-1})$ and the filter likelihood density $p(\mathbf{z}_k|D_k)$ are both Gaussian, which eventually leads to a Gaussian posterior density $p(\mathbf{x}_k|D_k)$. The Gaussian is the most convenient and widely used density function for the following reasons:

- It has many distinctive mathematical properties.
  - The Gaussian family is closed under linear transformation and conditioning.
  - Uncorrelated jointly Gaussian random variables are independent.
- It approximates many physical random phenomena by virtue of the central limit theorem of probability theory (see Sections 5.7 and 6.7 in [19] for more details).

Under the Gaussian approximation, the functional recursion of the Bayesian filter reduces to an algebraic recursion operating only on means and covariances of various conditional densities encountered in the time and the measurement updates.

A. Time Update

In the time update, the Bayesian filter computes the mean $\hat{\mathbf{x}}_{k|k-1}$ and the associated covariance $P_{k|k-1}$ of the Gaussian predictive density as follows:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}(\mathbf{x}_k|D_{k-1}), \tag{8}$$

where $\mathbb{E}$ is the statistical expectation operator. Substituting (1) into (8) yields

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}\big[\mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1}) + \mathbf{v}_{k-1}|D_{k-1}\big]. \tag{9}$$

Because $\mathbf{v}_{k-1}$ is assumed to be zero-mean and uncorrelated with the past measurements, we get

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbb{E}\big[\mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1})|D_{k-1}\big]\\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1})p(\mathbf{x}_{k-1}|D_{k-1})d\mathbf{x}_{k-1}\\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1})\\ &\quad \times\mathcal{N}(\mathbf{x}_{k-1};\hat{\mathbf{x}}_{k-1|k-1},P_{k-1|k-1})d\mathbf{x}_{k-1}, \end{aligned} \tag{10}$$

where $\mathcal{N}(.,.)$ is the conventional symbol for a Gaussian density. Similarly, we obtain the error covariance

$$\begin{aligned}P_{k|k-1} &= \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T|\mathbf{z}_{1:k-1}]\\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1})\mathbf{f}^T(\mathbf{x}_{k-1},\mathbf{u}_{k-1})\\ &\quad \times\mathcal{N}(\mathbf{x}_{k-1};\hat{\mathbf{x}}_{k-1|k-1},P_{k-1|k-1})d\mathbf{x}_{k-1}\\ &\quad -\hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^T_{k|k-1} + Q_{k-1}. \end{aligned} \tag{11}$$

B. Measurement Update

It is well known that the errors in the predicted measurements are zero-mean white sequences [2], [20]. Under the assumption that these errors can be well approximated by the Gaussian, we write the filter likelihood density

$$p(\mathbf{z}_k|D_{k-1}) = \mathcal{N}(\mathbf{z}_k;\hat{\mathbf{z}}_{k|k-1},P_{zz,k|k-1}), \tag{12}$$

where the predicted measurement

$$\hat{\mathbf{z}}_{k|k-1} = \int_{\mathbb{R}^{n_x}} \mathbf{h}(\mathbf{x}_k,\mathbf{u}_k)\mathcal{N}(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k-1},P_{k|k-1})d\mathbf{x}_k, \tag{13}$$

and the associated covariance

$$\begin{aligned}P_{zz,k|k-1} &= \int_{\mathbb{R}^{n_x}} \mathbf{h}(\mathbf{x}_k,\mathbf{u}_k)\mathbf{h}^T(\mathbf{x}_k,\mathbf{u}_k)\\ &\quad \mathcal{N}(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k-1},P_{k|k-1})d\mathbf{x}_k\\ &\quad -\hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1} + R_k. \end{aligned} \tag{14}$$

Hence, we write the conditional Gaussian density of the joint state and the measurement

$$p\Big(\ [\mathbf{x}_k^T \quad \mathbf{z}_k^T]^T|D_{k-1}\ \Big) = \left(\mathcal{N}\Big(\begin{array}{c}\hat{\mathbf{x}}_{k|k-1}\\ \hat{\mathbf{z}}_{k|k-1}\end{array}\Big),\right.\left.\left(\begin{array}{cc}P_{k|k-1} & P_{xz,k|k-1}\\ P^T_{xz,k|k-1} & P_{zz,k|k-1}\end{array}\right)\right), \tag{15}$$

where the cross-covariance

$$\begin{aligned}P_{xz,k|k-1} &= \int_{\mathbb{R}^{n_x}} \mathbf{x}_k\mathbf{h}^T(\mathbf{x}_k,\mathbf{u}_k)\\ &\quad \times\mathcal{N}(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k-1},P_{k|k-1})d\mathbf{x}_k\\ &\quad -\hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1}. \end{aligned} \tag{16}$$

On the receipt of a new measurement $\mathbf{z}_k$, the Bayesian filter computes the posterior density $p(\mathbf{x}_k|D_k)$ from (15) yielding

$$p(\mathbf{x}_k|D_k) = \mathcal{N}(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k},P_{k|k}), \tag{17}$$

where

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + W_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \tag{18}$$

$$P_{k|k} = P_{k|k-1} - W_k P_{zz,k|k-1} W_k^T \tag{19}$$

$$W_k = P_{xz,k|k-1}P^{-1}_{zz,k|k-1}. \tag{20}$$

If $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are linear functions of the state, the Bayesian filter under the Gaussian assumption reduces to the Kalman filter. Table I shows how quantities derived above are called in the Kalman filtering framework:

TABLE I

$P_{zz,k\|k-1}$ in (14)	Innovation Covariance
$(\mathbf{z}_k - \hat{\mathbf{z}}_{k\|k-1})$ in (18)	Innovation
W_k in (20)	Kalman gain

The signal-flow diagram in Fig. 2 summarizes the steps involved in the recursion cycle of the Bayesian filter. The heart of the Bayesian filter is therefore how to compute Gaussian weighted integrals whose integrands are all of the

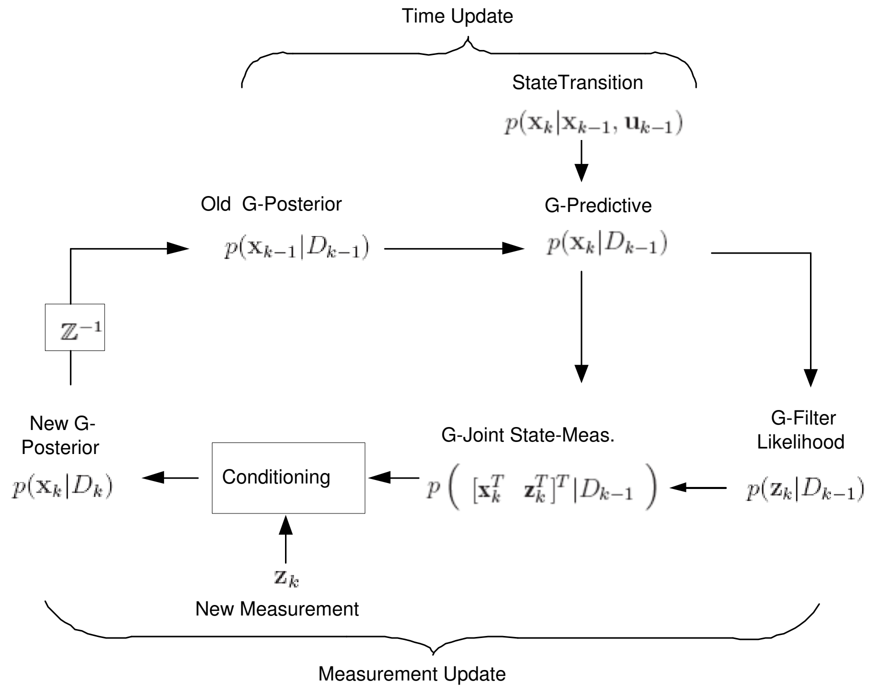


Fig. 2. Signal-flow diagram of the recursive Bayesian filter under the Gaussian assumption, where ‘G-’ stands for ’Gaussian-’.

form *nonlinear function* × *Gaussian density* that are present in (10)-(11), (13)-(14) and (16). The next section describes numerical integration methods to compute multi-dimensional weighted integrals.

III. A Review on numerical methods for Moment Integrals

Consider a multi-dimensional weighted integral of the form

$$I(\mathbf{f}) \quad = \quad \int_{\mathscr{D}} \mathbf{f}(\mathbf{x})w(\mathbf{x})d\mathbf{x}, \tag{21}$$

where $\mathbf{f}(.)$ is some arbitrary function, $\mathscr{D} \subseteq \mathbb{R}^n$ is the region of integration, and the known weighting function $w(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathscr{D}$. In a Gaussian-weighted integral, for example, $w(\mathbf{x})$ is a Gaussian density and satisfies the nonnegativity condition in the entire region. If the solution to the above integral (21) is difficult to obtain, we may seek numerical integration methods to compute it. The basic task of numerically computing the integral (21) is to find a set of points $\mathbf{x}_i$ and weights ω_i that approximates the integral $I(\mathbf{f})$ by a weighted sum of function evaluations:

$$I(\mathbf{f}) \quad \approx \quad \sum_{i=1}^{m} \omega_i \mathbf{f}(\mathbf{x}_i). \tag{22}$$

The methods used to find $\{\mathbf{x}_i, \omega_i\}$ can be divided into *product rules* and *non-product rules*, as described next.

A. Product Rules

For the simplest one-dimensional case (that is, $n = 1$), we may apply the quadrature rule to compute the integral (21) numerically [21], [22]. In the context of the Bayesian filter, we mention the Gauss-Hermite quadrature rule; when the weighting function $w(x)$ is in the form of a Gaussian density and the integrand $f(x)$ is well approximated by a polynomial in x, the Gauss-Hermite quadrature rule is used to compute the Gaussian-weighted integral efficiently [12].

The quadrature rule may be extended to compute multi-dimensional integrals by successively applying it in a tensor-product of one-dimensional integrals. Consider an m-point per dimension quadrature rule that is exact for polynomials of degree up to d. We set up a grid of m^n points for functional evaluations and numerically compute an n-dimensional integral while retaining the accuracy for polynomials of degree up to d only. Hence, the computational complexity of the product quadrature rule increases exponentially with n, and therefore suffers from the curse of dimensionality. Typically for $n > 5$, the product Gauss-Hermite quadrature rule is not a reasonable choice to approximate a recursive optimal Bayesian filter.

B. Non-Product Rules

To mitigate the curse of dimensionality issue in the product rules, we may seek non-product rules for integrals of arbitrary dimensions by choosing points directly from the domain of integration [18], [23]. Some of the well-known non-product rules include randomized Monte Carlo methods [4], quasi-Monte Carlo methods [24], [25], lattice rules [26] and sparse grids [27]–[29]. The randomized Monte Carlo methods evaluate the integration using a set of equally-weighted sample points drawn randomly, whereas in quasi-Monte Carlo methods and lattice rules the points are generated from a unit hyper-cube region using deterministically defined mechanisms. On the other hand, the sparse grids based on Smolyak formula in principle, combine a quadrature (univariate) routine for high-dimensional integrals more sophisticatedly; they detect important dimensions automatically and place more grid points there. Although the non-product methods mentioned here are powerful numerical integration tools to compute a given integral with a prescribed accuracy, they do suffer from the curse of dimensionality to certain extent [30].

C. The proposed method

In the recursive Bayesian estimation paradigm, we are interested in non-product rules that (i) yield reasonable accuracy, (ii) require small number of function evaluations and (iii) are easily extendable to arbitrarily high dimensions. In this paper we derive an efficient non-product cubature rule for Gaussian-weighted integrals. Specifically, we obtain a third-degree fully-symmetric cubature rule, whose complexity in terms of function evaluations increases linearly with the dimension n. Typically, a set of cubature points and weights are chosen so that the cubature rule is exact for a set of monomials of *degree* d or less, as shown by

$$\int_{\mathscr{D}} \mathscr{P}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} \quad = \quad \sum_{i=1}^{m} \omega_i \mathscr{P}(\mathbf{x}_i), \tag{23}$$

where $\mathscr{P}(\mathbf{x}) = x_1^{d_1} x_2^{d_2} \ldots x_n^{d_n}$; d_i are non-negative integers and $\sum_{i=1}^{n} d_i \leq d$. Here, an important quality criterion of a cubature rule is its degree; the higher the degree of the cubature rule is, the more accurate solution it yields. To find the unknowns $\{\mathbf{x}_i, \omega_i\}$ of the cubature rule of degree d, we solve a set of moment equations. However, solving the system of moment equations may be more tedious with increasing polynomial degree and/or dimension of the integration domain. For example, an m-point cubature rule entails $m(n+1)$ unknown parameters from its points and weights. In general, we may form a system of $\frac{(n+d)!}{n!d!}$ equations with respect to unknowns from distinct monomials of degree up to d. For the nonlinear system to have at least one solution (in this case, the system is said to be consistent), we use at least as many unknowns as equations [31]. That is, we choose m to be $m \geq \frac{(n+d)!}{(n+1)!d!}$. Suppose we obtain a cubature rule of degree three for $n = 20$. In this case, we solve $\frac{(20+3)!}{20!3!} = 1771$ nonlinear moment equations; the resulting rule may consist of more than 85 ($> \frac{(20+3)!}{21!3!}$) weighted cubature points.

To reduce the size of the system of algebraically independent equations or equivalently the number of cubature points markedly, Sobolev proposed the *invariant theory* in 1962 [32] (see also [31] and the references therein for a recent account of the invariant theory). The invariant theory, in principle, discusses how to restrict the structure of a cubature rule by exploiting symmetries of the region of integration and the weighting function. For example, integration regions such as the unit hypercube, the unit hypersphere, and the unit simplex exhibit symmetry. Hence, it is reasonable to look for cubature rules sharing the same symmetry. For the case considered above ($n = 20$ and $d = 3$), using the invariant theory, we may construct a cubature rule consisting of $2n(= 40)$ cubature points by solving only a pair of moment equations (see Section IV).

Note that the points and weights of the cubature rule are independent of the integrand $\mathbf{f}(\mathbf{x})$. Hence, they can be computed off-line and stored in advance to speed up the filter execution.

IV. The Cubature Kalman Filter

As described in Section II, nonlinear filtering in the Gaussian domain reduces to a problem of how to compute integrals, whose integrands are all of the form *nonlinear function×Gaussian density*. Specifically, we consider an integral of the form

$$I(\mathbf{f}) \quad = \quad \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x}) \text{exp}(-\mathbf{x}^T \mathbf{x}) d\mathbf{x} \tag{24}$$

defined in the Cartesian coordinate system. To compute the above integral numerically we take the following two steps: (i) We transform it into a more familiar spherical-radial integration form (ii) Subsequently, we propose a third-degree spherical-radial rule.

A. Transformation

In the spherical-radial transformation, the key step is a change of variable from the Cartesian vector $\mathbf{x} \in \mathbf{R}^n$ to a radius r and direction vector $\mathbf{y}$ as follows: Let $\mathbf{x} = r\mathbf{y}$ with $\mathbf{y}^T \mathbf{y} = 1$, so that $\mathbf{x}^T \mathbf{x} = r^2$ for $r \in [0, \infty)$. Then the integral (24) can be rewritten in a spherical-radial coordinate system as

$$I(\mathbf{f}) \quad = \quad \int_0^\infty \int_{U_n} \mathbf{f}(r\mathbf{y}) r^{n-1} \text{exp}(-r^2) d\sigma(\mathbf{y}) dr, \tag{25}$$

where U_n is the surface of the sphere defined by $U_n = \{\mathbf{y} \in \mathbb{R}^n | \ \mathbf{y}^T \mathbf{y} = 1\}$ and $\sigma(.)$ is the spherical surface measure or the area element on U_n. We may thus write the *radial integral*

$$I \quad = \quad \int_0^\infty S(r) r^{n-1} \text{exp}(-r^2) dr, \tag{26}$$

where $S(r)$ is defined by the *spherical integral* with the unit weighting function $w(\mathbf{y}) = 1$:

$$S(r) = \int_{U_n} \mathbf{f}(r\mathbf{y}) d\sigma(\mathbf{y}). \tag{27}$$

The spherical and the radial integrals are numerically computed by the spherical cubature rule (Subsection B below) and the Gaussian quadrature rule (Subsection C below), respectively. Before proceeding further, we introduce a number of notations and definitions when constructing such rules as follows:

- A cubature rule is said to be *fully symmetric* if the following two conditions hold:
  1) $\mathbf{x} \in \mathscr{D}$ implies $\mathbf{y} \in \mathscr{D}$, where $\mathbf{y}$ is any point obtainable from $\mathbf{x}$ by permutations and/or sign changes of the coordinates of $\mathbf{x}$.
  2) $w(\mathbf{x}) = w(\mathbf{y})$ on the region $\mathscr{D}$. That is, all points in the fully symmetric set yield the same weight value.

  For example, in the one-dimensional space, a point $x \in \mathbb{R}$ in the fully symmetric set implies that $(-x) \in \mathbb{R}$ and $w(x) = w(-x)$.
- In a fully symmetric region, we call a point $\mathbf{u}$ as a *generator* if $\mathbf{u} = (u_1, u_2, \ldots, u_r, 0, \ldots 0) \in \mathbb{R}^n$, where

$u_i \geq u_{i+1} > 0,\ \ i = 1, 2, \ldots (r-1)$. The new $\mathbf{u}$ should not be confused with the control input $\mathbf{u}$.

- For brevity, we suppress $(n-r)$ zero coordinates and use the notation $[u_1, u_2, \ldots u_r]$ to represent a complete fully symmetric set of points that can be obtained by permutating and changing the sign of the generator $\mathbf{u}$ in all possible ways. Of course, the complete set entails $\frac{2^r n!}{(n-r)!}$ points when $\{u_i\}$ are all distinct. For example, $[1] \in \mathbb{R}^2$ represents the following set of points:

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}.$$

Here, the generator is $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

- We use $[u_1, u_2, \ldots u_r]_i$ to denote the i-th point from the set $[u_1, u_2, \ldots u_r]$.

B. Spherical Cubature Rule

We first postulate a third-degree spherical cubature rule that takes the simplest structure due to the invariant theory:

$$\int_{U_n} \mathbf{f}(\mathbf{y}) d\sigma(\mathbf{y}) \approx \omega \sum_{i=1}^{2n} \mathbf{f}[u]_i. \tag{28}$$

The point set due to $[u]$ is invariant under permutations and sign changes. For the above choice of the rule (28), the monomials $\{y_1^{d_1} y_2^{d_2} \ldots y_n^{d_n}\}$ with $\sum_{i=1}^n d_i$ being an odd integer, are integrated exactly. In order that this rule is exact for all monomials of degree up to three, it remains to require that the rule be exact for all monomials for which $\sum_{i=1}^n d_i = 0, 2$. Equivalently, to find the unknown parameters u and ω, it suffices to consider monomials $\mathbf{f}(\mathbf{y}) = 1$, and $\mathbf{f}(\mathbf{y}) = y_1^2$ due to the fully symmetric cubature rule:

$$\mathbf{f}(\mathbf{y}) = 1: \qquad 2n\omega = \int_{U_n} d\sigma(\mathbf{y}) = A_n \tag{29}$$

$$\mathbf{f}(\mathbf{y}) = y_1^2: \qquad 2\omega u^2 = \int_{U_n} y_1^2 d\sigma(\mathbf{y}) = \frac{A_n}{n}, \tag{30}$$

where the surface area of the unit sphere $A_n = \frac{2\sqrt{\pi^n}}{\Gamma(n/2)}$ with $\Gamma(n) = \int_0^\infty x^{n-1} \exp(-x) dx$. Solving (29)-(30) yields $\omega = \frac{A_n}{2n}$, and $u^2 = 1$. Hence, the cubature points are located at the intersection of the unit sphere and its axes.

C. Radial Rule

We next propose a Gaussian quadrature for the radial integration. The Gaussian quadrature is known to be the most efficient numerical method to compute a one-dimensional integration [21], [22]. An m-point Gaussian quadrature is exact up to polynomials of degree $(2m-1)$ and constructed as follows:

$$\int_a^b f(x) w(x) dx \approx \sum_{i=1}^m \omega_i f(x_i), \tag{31}$$

where $w(x)$ is a known weighting function and non-negative on the interval $[a, b]$; the points $\{x_i\}$ and the associated weights $\{\omega_i\}$ are unknowns to be determined uniquely. In our case, a comparison of (26) and (31) yields the weighting function and the interval to be $w(x) = x^{n-1}\exp(-x^2)$ and $[0, \infty)$, respectively. To transform this integral into an integral for which the solution is familiar, we make another change of variable via $t = x^2$ yielding

$$\int_0^\infty f(x) x^{n-1} \exp(-x^2) dx = \frac{1}{2} \int_0^\infty \tilde{f}(t) t^{(\frac{n}{2}-1)} \times \exp(-t) dt, \tag{32}$$

where $\tilde{f}(t) = f(\sqrt{t})$. The integral on the right-hand side of (32) is now in the form of the well-known generalized Gauss-Laguerre formula. The points and weights for the generalized Gauss-Laguerre quadrature are readily obtained as discussed elsewhere [21]. A first-degree Gauss-Laguerre rule is exact for $\tilde{f}(t) = 1, t$. Equivalently, the rule is exact for $f(x) = 1, x^2$; it is not exact for odd degree polynomials such as $f(x) = x, x^3$. Fortunately, when the radial-rule is combined with the spherical rule to compute the integral (24), the (combined) spherical-radial rule vanishes for all odd-degree polynomials; the reason is that the spherical rule vanishes by symmetry for any odd-degree polynomial (see (25)). Hence, the spherical-radial rule for (24) is exact for all odd degrees. Following this argument, for a spherical-radial rule to be exact for all third-degree polynomials in $\mathbf{x} \in \mathbb{R}^n$, it suffices to consider the first-degree generalized Gauss-Laguerre rule entailing a single point and weight. We may thus write

$$\int_0^\infty f(x) x^{n-1} \exp(-x^2) dx \approx \omega_1 f(x_1), \tag{33}$$

where the point x_1 is chosen to be the square-root of the root of the first-order generalized Laguerre polynomial, which is orthogonal with respect to the modified weighting function $x^{(\frac{n}{2}-1)}\exp(-x)$; subsequently, we find ω_1 by solving the zeroth-order moment equation appropriately. In this case, we have $\omega_1 = \frac{\Gamma(n/2)}{2}$, and $x_1 = \sqrt{\frac{n}{2}}$. A detailed account of computing the points and weights of a Gaussian quadrature with the classical and nonclassical weighting function is presented in [33].

D. Spherical-Radial Rule

In this subsection, we describe two useful results that are used to (i) combine the spherical and radial rule obtained separately, and (ii) extend the spherical-radial rule for a Gaussian weighted integral. The respective results are presented as two theorems:

Theorem 4.1: *Let the radial integral be computed numerically by the m_r-point Gaussian quadrature rule*

$$\int_0^\infty f(r) r^{n-1} \exp(-r^2) dr = \sum_{i=1}^{m_r} a_i f(r_i).$$

Let the spherical integral be computed numerically by the m_s-point spherical rule

$$\int_{U_n} \mathbf{f}(r\mathbf{s}) d\sigma(\mathbf{s}) = \sum_{j=1}^{m_s} b_j \mathbf{f}(r\mathbf{s}_j).$$

Then, an $(m_s \times m_r)$*-point spherical-radial cubature rule is given by*

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\text{exp}(-\mathbf{x}^T\mathbf{x})d\mathbf{x} \quad \approx \quad \sum_{j=1}^{m_s}\sum_{i=1}^{m_r} a_i b_j \mathbf{f}(r_i \mathbf{s}_j). \quad (34)$$

Proof: Because cubature rules are devised to be exact for a subspace of monomials of some degree, we consider an integrand of the form

$$\mathbf{f}(\mathbf{x}) \quad = \quad x_1^{d_1} x_2^{d_2} \ldots x_n^{d_n},$$

where $\{d_i\}$ are some positive integers. Hence, we write the integral of interest

$$I(\mathbf{f}) \quad = \quad \int_{\mathbb{R}^n} x_1^{d_1} x_2^{d_2} \ldots x_n^{d_n} \text{exp}(-\mathbf{x}^T\mathbf{x})d\mathbf{x}.$$

For the moment, we assume the above integrand to be a monomial of degree d exactly; that is, $\sum_{i=1}^n d_i = d$. Making the change of variable as described in Subsection A, we get

$$\begin{aligned} I(\mathbf{f}) \quad &= \quad \int_0^\infty \int_{U_n} (ry_1)^{d_1}(ry_2)^{d_2} \ldots (ry_n)^{d_n} r^{n-1} \\ &\qquad \times \text{exp}(-r^2) d\sigma(\mathbf{y}) dr. \end{aligned}$$

Decomposing the above integration into the radial and spherical integrals yields

$$I(\mathbf{f}) \quad = \quad \int_0^\infty r^{n+d-1}\text{exp}(-r^2)dr \int_{U_n} y_1^{d_1} y_2^{d_2} \ldots y_n^{d_n} d\sigma(\mathbf{y}).$$

Applying the numerical rules appropriately, we have

$$\begin{aligned} I(\mathbf{f}) \quad &\approx \quad \Big(\sum_{i=1}^{m_r} a_i r_i^d\Big)\Big(\sum_{j=1}^{m_s} b_j s_{j1}^{d_1} s_{j2}^{d_2} \ldots s_{jn}^{d_n}\Big) \\ &= \quad \sum_{i=1}^{m_r}\sum_{j=1}^{m_s} a_i b_j (r_i s_{j1})^{d_1}(r_i s_{j2})^{d_2} \ldots (r_i s_{jn})^{d_n}, \end{aligned}$$

as desired. As we may extend the above results for monomials of degree less than d, the theorem holds for any arbitrary integrand that can be written as a linear combination of monomials of degree up to d (see also Section 2.8 in [18]). □

Theorem 4.2: *Let the weighting functions* $w_1(\mathbf{x})$ and $w_2(\mathbf{x})$ be $w_1(\mathbf{x}) = \text{exp}(-\mathbf{x}^T\mathbf{x})$ *and* $w_2(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mu, \Sigma)$. *Then for every square matrix* $\sqrt{\Sigma}$ *such that* $\sqrt{\Sigma}\sqrt{\Sigma}^T = \Sigma$, we have

$$\begin{aligned} \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x}) w_2(\mathbf{x}) d\mathbf{x} \quad &= \quad \frac{1}{\sqrt{\pi^n}} \int_{\mathbb{R}^n} \mathbf{f}(\sqrt{2\Sigma}\mathbf{x} + \mu) \\ &\qquad \times w_1(\mathbf{x}) d\mathbf{x}. \qquad (35) \end{aligned}$$

Proof: Consider the left-hand side of (35). Because Σ is a positive definite matrix, we factorize Σ to be $\Sigma = \sqrt{\Sigma}\sqrt{\Sigma}^T$. Making a change of variable via $\mathbf{x} = \sqrt{2\Sigma}\mathbf{y} + \mu$, we get

$$\begin{aligned} \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x}; \mu, \Sigma)d\mathbf{x} \quad &= \quad \int_{\mathbb{R}^n} \mathbf{f}(\sqrt{2\Sigma}\mathbf{y} + \mu)\frac{1}{\sqrt{|2\pi\Sigma|}} \\ &\qquad \times \text{exp}(-\mathbf{y}^T\mathbf{y})|\sqrt{2\Sigma}|d\mathbf{y} \\ &= \quad \frac{1}{\sqrt{\pi^n}} \int_{\mathbb{R}^n} \mathbf{f}(\sqrt{2\Sigma}\mathbf{y} + \mu) w_1(\mathbf{y}) d\mathbf{y} \\ &= \quad \frac{1}{\sqrt{\pi^n}} \int_{\mathbb{R}^n} \mathbf{f}(\sqrt{2\Sigma}\mathbf{x} + \mu) w_1(\mathbf{x}) d\mathbf{x} \end{aligned}$$

which proves the theorem. □

For the third-degree spherical-radial rule, $m_r = 1$ and $m_s = 2n$. Hence, it entails a total of $2n$ cubature points. Using the above theorems, we extend this third-degree spherical-radial rule to compute a standard Gaussian weighted integral as follows:

$$I_{\mathcal{N}}(\mathbf{f}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{I})d\mathbf{x} \quad \approx \quad \sum_{i=1}^{m} \omega_i \mathbf{f}(\xi_i),$$

where

$$\begin{aligned} \xi_i \quad &= \quad \sqrt{\frac{m}{2}}[1]_i \\ \omega_i \quad &= \quad \frac{1}{m}, \qquad i = 1, 2, \ldots m = 2n. \end{aligned}$$

We use the cubature-point set $\{\xi_i, \omega_i\}$ to numerically compute integrals (10)-(11), and (13)-(16) and obtain the CKF algorithm, details of which are presented in Appendix A. Note that the above cubature-point set is now defined in the Cartesian coordinate system.

V. Is there a need for higher-degree cubature rules ?

In this section, we emphasize the importance of third-degree cubature rules over higher-degree rules (degree more than three), when they are embedded into the cubature Kalman filtering framework for the following reasons:

- *Sufficient approximation.* The CKF recursively propagates the first two-order moments, namely, the mean and covariance of the state variable. A third-degree cubature rule is also constructed using up to the second-order moment.

  Moreover, a natural assumption for a nonlinearly transformed variable to be closed in the Gaussian domain is that the nonlinear function involved is reasonably smooth. In this case, it may be reasonable to assume that the given nonlinear function can be well-approximated by a quadratic function near the prior mean. Because the third-degree rule is exact up to third-degree polynomials, it computes the posterior mean accurately in this case. However, it computes the error covariance approximately; for the covariance estimate to be more accurate, a cubature rule is required to be exact at least up to a fourth degree polynomial. Nevertheless, a higher-degree rule will translate to higher accuracy only if the integrand is well-behaved in the sense of being approximated by a higher-degree polynomial, and the weighting function is known to be a Gaussian density exactly. In practice, these two requirements are hardly met. *However, considering in* *the cubature Kalman filtering framework, our experience with higher-degree rules has indicated that they yield no improvement or make the performance worse.*

- *Efficient and robust computation.* The theoretical lower bound for the number of cubature points of a third-degree centrally symmetric cubature rule is given by twice the dimension of an integration region [34]. Hence, the proposed spherical-radial cubature rule is considered to be the most efficient third-degree cubature rule. Because the number of points or function evaluations in the proposed cubature rules scales linearly with the dimension, it may be considered as a practical step for easing the curse of dimensionality.

  According to [35] and Section 1.5 in [18], a 'good' cubature rule has the following two properties: (i) all the cubature points lie inside the region of integration, and (ii) all the cubature weights are positive. The proposed rule entails $2n$ equal, positive weights for an n-dimensional unbounded region and hence belongs to a good cubature family. Of course, we hardly find higher-degree cubature rules belonging to a good cubature family especially for high-dimensional integrations.

In the final analysis, the use of higher-degree cubature rules in the design of the CKF may often sabotage its performance.

VI. The Square-Root Cubature Kalman Filter

This section addresses (i) the rationale for why we need a square-root extension of the standard CKF and (ii) how the square-root solution can be developed systematically. The two basic properties of an error covariance matrix are (i) symmetry and (ii) positive definiteness. It is important that we preserve these two properties in each update cycle. The reason is that the use of a forced symmetry on the solution of the matrix Ricatti equation improves the numerical stability of the Kalman filter [36], whereas the underlying meaning of the covariance is embedded in the positive definiteness. In practice, due to errors introduced by arithmetic operations performed on finite word-length digital computers, these two properties are often lost. Specifically, the loss of the positive definiteness may probably be more hazardous as it stops the CKF to run continuously. In each update cycle of the CKF, we mention the following numerically sensitive operations that may catalyze to destroy the properties of the covariance:

- Matrix square-rooting (see (38) and (43)).
- Matrix inversion (see (49)).
- Matrix squared-form amplifying roundoff errors (see (42), (47) and (48)).
- Substraction of the two positive definite matrices present in the covariant update (see (51)).

Moreover, some nonlinear filtering problems may be numerically ill. For example, the covariance is likely to turn out to be non-positive definite when (i) very accurate measurements are processed, or (ii) a linear combination of state vector components is known with greater accuracy while other combinations are essentially unobservable [37].

As a systematic solution to mitigate ill effects that may eventually lead to an unstable or even divergent behavior, the logical procedure is to go for a square-root version of the CKF, hereafter called *square-root cubature Kalman filter* (SCKF). The SCKF essentially propagates square-root factors of the predictive and posterior error covariances. Hence, we avoid matrix square-rooting operations. In addition, the SCKF offers the following benefits [38]:

- Preservation of symmetry and positive (semi)definiteness of the covariance.
- Improved numerical accuracy owing to the fact that $\kappa(S) = \sqrt{\kappa(S^T S)}$, where the symbol κ denotes the condition number.
- Doubled-order precision.

To develop the SCKF, we use (i) the least-squares method for the Kalman gain and (ii) matrix triangular factorizations or triangularizations (e.g., the QR decomposition) for covariance updates. The least-squares method avoids to compute a matrix inversion explicitly, whereas the triangularization essentially computes a triangular square-root factor of the covariance without square-rooting a squared-matrix form of the covariance. Appendix B presents the SCKF algorithm, where all of the steps can be deduced directly from the CKF except for the update of the posterior error covariance; hence we derive it in a squared-equivalent form of the covariance in the appendix.

The computational complexity of the SCKF in terms of flops, grows as the cube of the state dimension, hence it is comparable to that of the CKF or the EKF. We may reduce the complexity significantly by (i) manipulating sparsity of the square-root covariance carefully and (ii) coding triangularization algorithms for distributed processor-memory architectures.

VII. A Comparison of UKF with CKF

Similarly to the CKF, the unscented Kalman filter (UKF) is another approximate Bayesian filter built in the Gaussian domain, but uses a completely different set of deterministic weighted points [10], [39]. To elaborate the approach taken in the UKF, consider an n-dimensional random variable $\mathbf{x}$ having a symmetric prior density $\Pi(\mathbf{x})$ with mean μ and covariance Σ, within which the Gaussian is a special case. Then a set of $(2n+1)$ sample points and weights, $\{\mathcal{X}_i, \omega_i\}_{i=0}^{2n}$ are chosen to satisfy the following moment-matching conditions:

$$\begin{aligned} \sum_{i=0}^{2n} \omega_i \mathcal{X}_i &= \mu \\ \sum_{i=0}^{2n} \omega_i \big(\mathcal{X}_i - \mu_x\big)\big(\mathcal{X}_i - \mu_x\big)^T &= \Sigma. \end{aligned}$$

Among many candidate sets, one symmetrically distributed sample point set, hereafter called the *sigma-point set*, is picked up as follows:

$$\begin{aligned} \mathcal{X}_0 = \mu, \qquad \omega_0 &= \frac{\kappa}{n+\kappa} \\ \mathcal{X}_i = \mu + (\sqrt{(n+\kappa)\Sigma})_i, \qquad \omega_i &= \frac{1}{2(n+\kappa)} \\ \mathcal{X}_{n+i} = \mu - (\sqrt{(n+\kappa)\Sigma})_i, \qquad \omega_{n+i} &= \frac{1}{2(n+\kappa)}, \end{aligned}$$

where $i = 1, 2, \ldots n$ and the i-th column of a matrix A is denoted by $(A)_i$; the parameter κ is used to scale the spread of sigma points from the prior mean μ, hence the name "scaling parameter". Due to its symmetry, the sigma-point set matches the skewness. Moreover, to capture the kurtosis of the prior density closely, it is suggested that we choose κ to be $\kappa = (3-n)$ (Appendix I of [10], [39]). This choice preserves moments up to the fifth order exactly in the simple one-dimensional Gaussian case.

In summary, the sigma-point set is chosen to capture a number of low-order moments of the prior density $p(\mathbf{x})$ as correctly as possible.

Then the *unscented transformation* is introduced as a method of computing posterior statistics of $\mathbf{y} \in \mathbf{R}^m$ that are related to $\mathbf{x}$ by a nonlinear transformation $\mathbf{y} = \mathbf{f}(\mathbf{x})$. It approximates the mean and the covariance of $\mathbf{y}$ by a weighted sum of projected sigma points in the $\mathbf{R}^m$ space, as shown by

$$\begin{aligned} \mathbb{E}[\mathbf{y}] &= \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\Pi(\mathbf{x})d\mathbf{x} \\ &\approx \sum_{i=0}^{2n} \omega_i \mathcal{Y}_i \end{aligned} \tag{36}$$

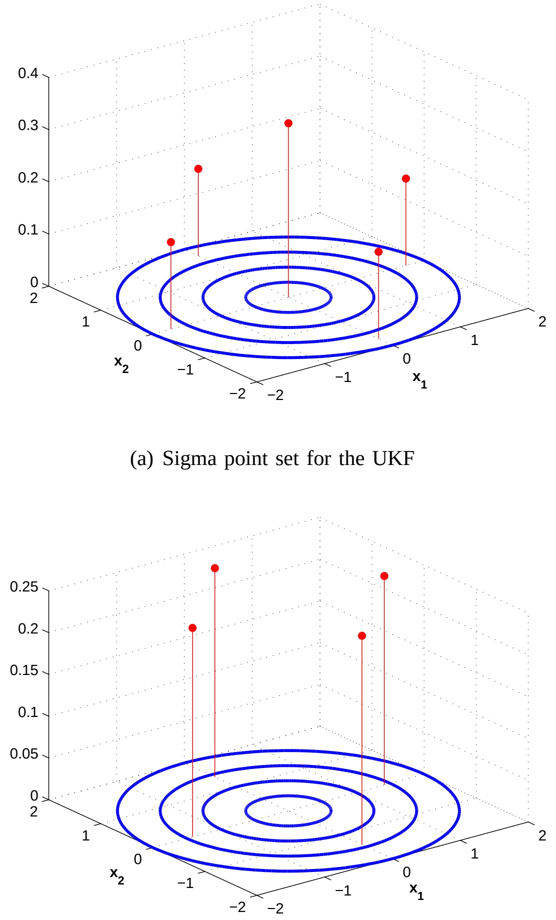
$$\begin{aligned} \text{cov}[\mathbf{y}] &= \int_{\mathbb{R}^n} (\mathbf{f}(\mathbf{x}) - \mathbb{E}[\mathbf{y}])(\mathbf{f}(\mathbf{x}) - \mathbb{E}[\mathbf{y}])^T \Pi(\mathbf{x})d\mathbf{x} \\ &\approx \sum_{i=0}^{2n} \omega_i (\mathcal{Y}_i - \mathbb{E}[\mathbf{y}])(\mathcal{Y}_i - \mathbb{E}[\mathbf{y}])^T, \end{aligned} \tag{37}$$

where $\mathcal{Y}_i = \mathbf{f}(\mathcal{X}_i),\ i = 0, 1, \ldots 2n$. The unscented transformation approximating the mean and the covariance, in particular, is correct up to a p-th order nonlinearity when the sigma-point set correctly captures the first p order prior moments. This can be proved by comparing the Taylor expansion of the nonlinear function $\mathbf{f}(\mathbf{x})$ up to p-order terms and the statistics computed by the unscented transformation; here, $\mathbf{f}(\mathbf{x})$ is expanded about the true (prior) mean μ, which is related to $\mathbf{x}$ via $\mathbf{x} = \mu + \mathbf{e}$ with the perturbation error $\mathbf{e} \sim \mathcal{N}(0, \Sigma)$ [10].

The UKF and the CKF share a common property- they use a weighted set of symmetric points. Fig. 3, shows the spread of the weighted sigma-point set and the proposed cubature-point set, respectively in the two-dimensional space; the points and their weights are denoted by the location and the height of the stems, respectively. However, as shown in Fig. 3, they are fundamentally different. To elaborate further, we derive the cubature-point set built into the new CKF with totally different philosophy in mind:

- We assume the prior statistics of $\mathbf{x}$ to be Gaussian instead of a more general symmetric density.
- Subsequently, we focus on how to compute the posterior statistics of $\mathbf{y}$ accurately. To be more specific, we need to compute the first two-order moments of $\mathbf{y}$ exactly because we embed them in a linear update rule.

In this line of thinking, the solution to the problem at hand boils down to the efficient third-order cubature rule. As in the sigma-point approach, we do not treat the derivation of a point set for the prior density and the computation of posterior statistics as two disjoint problems.



(a) Sigma point set for the UKF

(b) Third-degree spherical-radial cubature point set for the CKF

Fig. 3. Two kinds of distributions of point sets in two-dimensional space.

Moreover, suppose a given function $\mathbf{f}(.)$ is a linear combination of a monomial of degree at most three and some other higher odd-degree monomials. Then we may readily guarantee that the error, incurred in computing the integral of $\mathbf{f}(.)$ with respect to a Gaussian weight function, vanishes. As in the sigma-point approach, the function $\mathbf{f}(.)$ does not need to be well approximated by the Taylor polynomial about a prior mean.

Finally, we mention the following limitations of the sigma-point set built into the UKF, which are not present in the cubature-point set built into the CKF:

- *Numerical inaccuracy.* Traditionally, there has been more emphasis on cubature rules having desirable numerical quality criterion than on the efficiency. It is proven that the cubature rule implemented in a finite-precision arithmetic machine introduces a large amount of roundoff errors when the stability factor defined by $\frac{\sum_i |\omega_i|}{\sum_i \omega_i}$, is larger than unity [18], [28]. We look at formulas (36)-(37) in the unscented transformation from the numerical integration perspective. In this case, when n goes beyond three, $\sum_{i=0}^{2n} |\omega_i| = (\frac{2n}{3} - 1)$ and $\sum_{i=0}^{2n} \omega_i = 1$. Hence, the stability factor scales linearly with n, thereby inducing significant perturbations in numerical estimates for

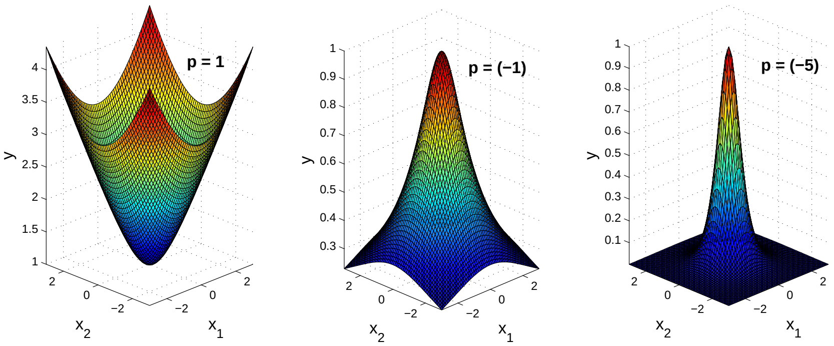


Fig. 4. Effect of the power p on the shape of $y = (\sqrt{1 + \mathbf{x}^T\mathbf{x}})^p$.

moment integrals.

- *Unavailability of a square-root solution.* We perform the square-rooting operation (or the Cholesky factorization) on the error covariance matrix as the first step of both the time and measurement updates in each cycle of both the UKF and the CKF. From the implementation point of view, the square-rooting is one of the most numerically sensitive operations. To avoid this operation in a systematic manner, we may seek to develop a square-root version of the UKF. Unfortunately, it may be impossible for us to formulate the square-root UKF that enjoys numerical advantages similar to the SCKF. The reason is that when a negatively weighted sigma point is used to update any matrix, the resulting down-dated matrix may possibly be non-positive definite. Hence, errors may occur when executing a 'pseudo' square-root version of the UKF in a limited word-length system (see the pseudo square-root version of the UKF in [40]).
- *Filter instability.* Given no computational errors due to an arithmetic imprecision, the presence of the negative weight may still prohibit us from writing a covariance matrix P in a squared form such that $P = SS^T$. To state it in another way, the UKF-computed covariance matrix is not always guaranteed to be positive definite. Hence, the unavailability of the square-root covariance causes the UKF to halt its operation. This could be disastrous in practical terms. To improve stability, heuristic solutions such as fudging the covariance matrix artificially (Appendix III of [10]) and the scaled unscented transformation [41] are proposed.

Note that the parameter κ of the sigma-point set may be forced to be zero to match the cubature-point set. Unfortunately, in the past there has been no mathematical justification or motivation for choosing $\kappa = 0$ for its associated interpretabilities. Instead, more secondary scaling parameters (apart from κ) were introduced in an attempt to incorporate knowledge about the prior statistics in the unscented transformation [41].

To sum up, we claim that the cubature approach is more accurate and more principled in mathematical terms than the sigma-point approach.

VIII. Simulations

In this section, we report the experimental results obtained by applying the CKF when applied to two nonlinear state-estimation problems: In the first high-dimensional illustrative experiment, we use the proposed cubature rules to estimate the mean and variance of a nonlinearly transformed Gaussian random variable. In the second application-oriented experiment, we use the CKF to track a maneuvering aircraft in two different setups- in the first setup, we assume the measurement noise to be Gaussian, whereas in the second setup, it takes a Gaussian mixture model.

A. Illustrative high-dimensional example

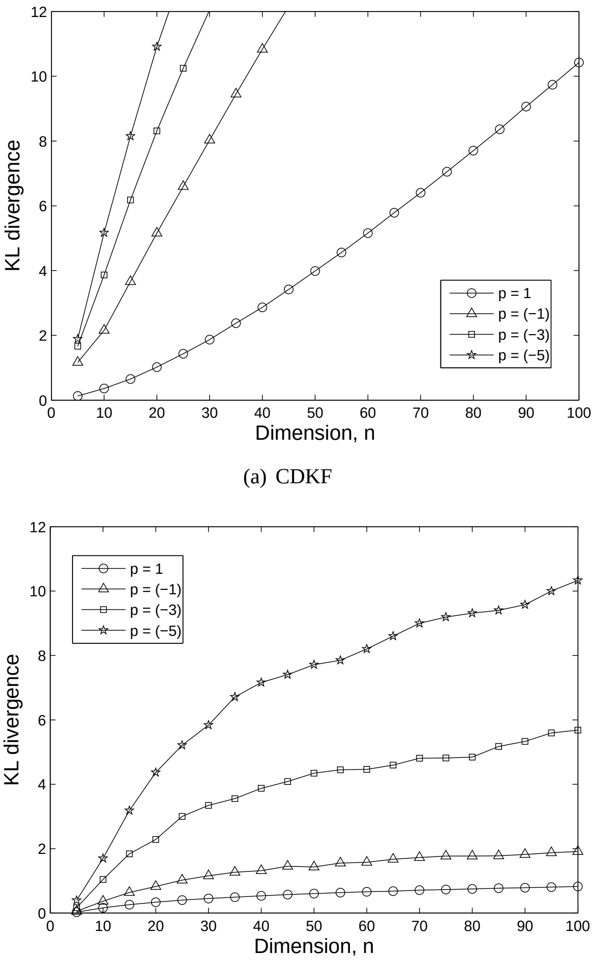
As described in Section II, the CKF uses the spherical-radial cubature rule to numerically compute the moment integrals of the form *nonlinear function* × *Gaussian*. The more accurate the estimate is, the quicker the filter converges to the correct state. We consider a general form of the multi-quadric function

$$y \;=\; \left(\sqrt{1 + \mathbf{x}^T\mathbf{x}}\right)^p$$

where $\mathbf{x}$ is assumed to be an n-dimensional Gaussian random variable with mean μ, and covariance Σ. In this experiment, p takes values $p = 1, -1, -3$, and -5 (Fig. 4). Our objective is to use four different methods, namely (i) the first-order Taylor series-based approximation built into the EKF, (ii) the unscented transformation (with $\kappa = 3 - n$) built into the UKF, (iii) the cental difference method with a step-size of $\sqrt{3}$ built into the CDKF and (iv) the spherical-radial rule built into the CKF, to compute the first two order (uncentralized) moments

$$\begin{aligned} \mathbb{E}(y) &= \int_{\mathbb{R}^n} (\sqrt{1 + \mathbf{x}^T\mathbf{x}})^p \mathcal{N}(\mathbf{x}; \mu, \Sigma) d\mathbf{x} \\ \text{and } \mathbb{E}(y^2) &= \int_{\mathbb{R}^n} (1 + \mathbf{x}^T\mathbf{x})^p \mathcal{N}(\mathbf{x}; \mu, \Sigma) d\mathbf{x} \end{aligned}$$

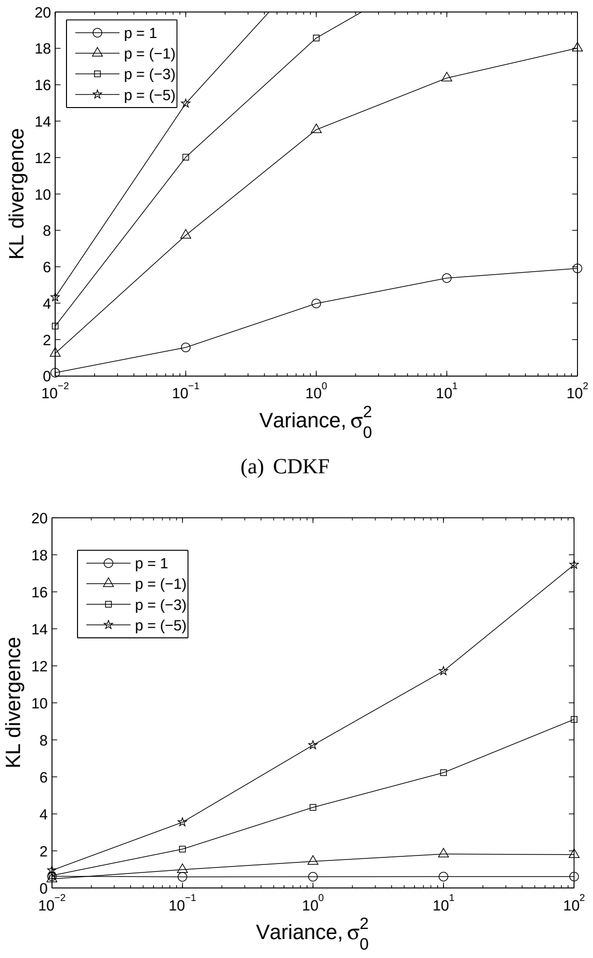
for different dimensions and prior information. We set μ to be zero because the function has a significant nonlinearity near the origin. Covariance matrices are randomly generated with diagonal entries being all equal to σ_0^2 for 50 independent runs. The Monte Carlo (MC) method with 10,000 samples is used to obtain the optimal estimate. To compare the filter estimated statistics against the optimal statistics, we use the Kullback-Leibler (KL) divergence assuming the statistics are Gaussian.



(a) CDKF

(b) CKF

Fig. 5. Ensemble-averaged KL divergence plots when $\sigma_0^2 = 1$.



(a) CDKF

(b) CKF

Fig. 6. Ensemble-averaged KL divergence plots when $n = 50$.

Of course, the smaller the KL divergence is, the better the filter estimate is.

Figs. 5(a) and 5(b) show how the KL divergence of the CDKF and CKF estimates varies, respectively when the dimension n increases for $\sigma_0^2 = 1$. The EKF computes a zero variance due to zero gradient at the prior mean; the unscented transformation often fails to yield meaningful results due to the numerical ills identified in Section VII; hence, we do not present the results of the EKF and UKF. As shown in Figs. 5(a) and 5(b), the CDKF and the CKF estimates degrade when the dimension n increases. The reason is that the n-dimensional state vector is estimated from a scalar measurement, irrespective of n. Moreover, when the nonlinearity of the multi-quadric function becomes more 'severe' (when p changes from 1 to -5), we also see the degradation as expected. However, for all cases of p considered herein, the CDKF estimate is inferior to the CKF and the reason is as follows: Consider the integrand obtained from the product of a multi-quadric function and the standard Gaussian density over the radial variable. It has a peak occurring at distance from the origin proportional to $\sqrt{n}$. The central-difference method approximates the nonlinear function by a set of grid-points located at a fixed distance irrespective of n, thereby failing to capture global properties of the function. On the other hand, the cubature points are located at a radius that scales with $\sqrt{n}$. Note also that the third-degree spherical-radial rule is exact for polynomials of all odd degrees even beyond three (due to the symmetry of the spherical rule) and this may be another reason for the increased accuracy obtained using the CKF.

Figs. 6(a) and 6(b) show how the KL divergence of the CDKF and CKF estimates varies, respectively when the prior variance σ_0^2 increases for a dimension $n = 50$. As expected, an increase in the prior variance σ_0^2 causes the posterior variance to increase. Hence, all filter estimates degrade or the KL divergence increases when we increase σ_0^2, as shown in Fig. 6. However, for all values of p, again we find that the CKF estimate is superior to the CDKF and the reason can be attributed to the facts just mentioned. The CDKF estimate is seen to be more degraded and deviate significantly from the CKF estimate when the prior variance is larger and the dimension is higher.

B. *Target tracking*

Scenario 1. We consider a typical air-traffic control scenario, where an aircraft executes maneuvering turn in a horizontal

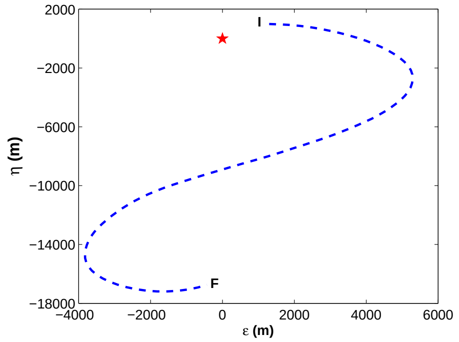


Fig. 7. Aircraft trajectory (I - initial point, F - final point), ★ - Radar location.

plane at a constant, but unknown turn rate Ω. Fig. 7 shows a representative trajectory of the aircraft. The kinematics of the turning motion can be modeled by the following nonlinear process equation [42]:

$$\mathbf{x}_k = \begin{pmatrix} 1 & \frac{\sin\Omega T}{\Omega} & 0 & -\left(\frac{1-\cos\Omega T}{\Omega}\right) & 0 \\ 0 & \cos\Omega T & 0 & -\sin\Omega T & 0 \\ 0 & \frac{1-\cos\Omega T}{\Omega} & 1 & \frac{\sin\Omega T}{\Omega} & 0 \\ 0 & \sin\Omega T & 0 & \cos\Omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_{k-1} + \mathbf{v}_k,$$

where the state of the aircraft $\mathbf{x} = [\xi \;\; \dot{\xi} \;\; \eta \;\; \dot{\eta} \;\; \Omega]^T$; ξ and η denote positions, and $\dot{\xi}$ and $\dot{\eta}$ denote velocities in the x and y directions, respectively; T is the time-interval between two consecutive measurements; the process noise $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, Q)$ with a nonsingular covariance $Q = \text{diag}[q_1 M \;\; q_1 M \;\; q_2 T]$, where

$$M = \begin{pmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{pmatrix};$$

The scalar parameters q_1 and q_2 are related to process noise intensities. A radar is fixed at the origin of the plane and equipped to measure the range, r, and bearing, θ. Hence, we write the measurement equation

$$\begin{pmatrix} r_k \\ \theta_k \end{pmatrix} = \begin{pmatrix} \sqrt{\xi_k^2 + \eta_k^2} \\ \tan^{-1}(\frac{\eta_k}{\xi_k}) \end{pmatrix} + \mathbf{w}_k,$$

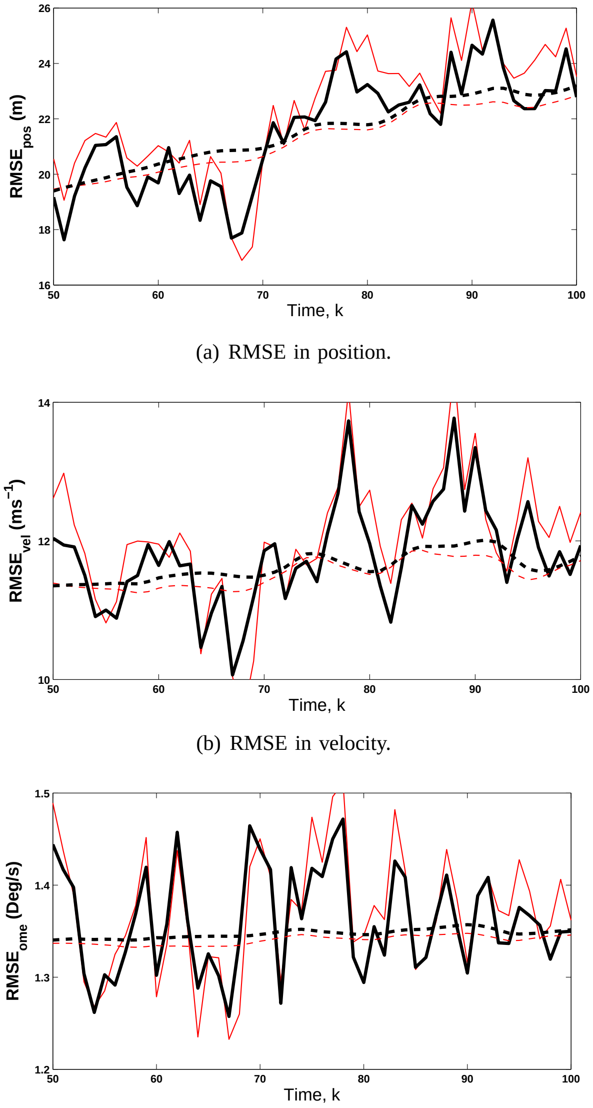
where the measurement noise $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, R)$ with $R = \text{diag}[\sigma_r^2 \;\; \sigma_\theta^2]$.

Data:

$$\begin{aligned} T &= 1\text{s} \\ \Omega &= -3^o\text{s}^{-1} \\ q_1 &= 0.1\text{m}^2\text{s}^{-3} \\ q_2 &= 1.75 \times 10^{-4}\text{s}^{-3} \\ \sigma_r &= 10\text{m} \\ \sigma_\theta &= \sqrt{10}\text{mrad} \end{aligned}$$

True initial state

$$\mathbf{x}_0 = [1000\text{m} \; 300\text{ms}^{-1} \; 1000\text{m} \; 0\text{ms}^{-1} \; -3^o\text{s}^{-1}]^T,$$



(a) RMSE in position.

(b) RMSE in velocity.

(c) RMSE in turn rate.

Fig. 8. True RMSE (solid thin- CDKF, solid thick- CKF) Vs. filter-estimated RMSE (dashed thin- CDKF, dashed thick- CKF) for the first scenario.

and the associated covariance

$$P_{0/0} = \text{diag}[100\text{m}^2 \; 10\text{m}^2\text{s}^{-2} \; 100\text{m}^2 \; 10\text{m}^2\text{s}^{-2} \; 100\text{mrad}^2\text{s}^{-2}].$$

The initial state estimate $\hat{\mathbf{x}}_{0/0}$ is chosen randomly from $\mathcal{N}(\mathbf{x}_0, P_{0/0})$ in each run; the total number of scans per run is 100.

To track the maneuvering aircraft we use the new square-root version of the CKF for its numerical stability and compare its performance against the EKF, the UKF, and the CDKF. For a fair comparison, we make 250 independent Monte Carlo runs. All the filters are initialized with the same condition in each run.

Performance metrics. To compare various nonlinear filter performances, we use the root-mean square error (RMSE) of the position, velocity and turn rate. The RMSE yields a combined measure of the bias and variance of a filter estimate.

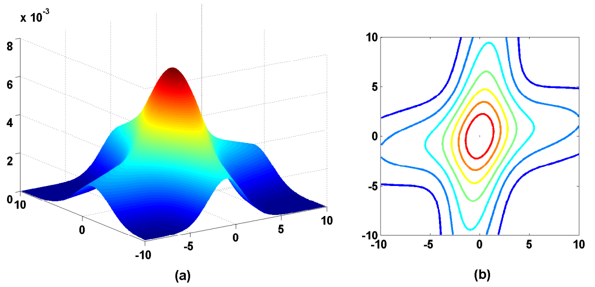


Fig. 9. Measurement noise: Gaussian mixture model and its contour plot.

We define the RMSE in position at time k, as

$$\text{RMSE}_{\text{pos}}(k) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left((\xi_k^n - \hat{\xi}_k^n)^2 + (\eta_k^n - \hat{\eta}_k^n)^2\right)},$$

where (ξ_k^n, η_k^n) and $(\hat{\xi}_k^n, \hat{\eta}_k^n)$ are the true and estimated positions at the n-th Monte Carlo run. Similarly to the RMSE in position, we may also write formulas of the RMSE in velocity and turn rate. As a self-assessment of its estimation errors, a filter provides an error covariance. Hence, we consider the filter-estimated RMSE as the square-root of the averaged (over Monte Carlo runs) appropriate diagonal entries of the covariance. The filter estimate is refereed to be consistent if the (true) RMSE is equal to its estimated RMSE. *We may expect the CKF to be inconsistent or divergent when its design assumption that the conditional densities are all Gaussian is violated.*

Figs. 8(a), 8(b) and 8(c) show the true and estimated RMSEs in position, velocity and turn rate, respectively for the CDKF and the CKF in an interval of 50-100s. Owing to the fact that the EKF is sensitive to information available in the previous time step, it diverges abruptly after 70-80s. The UKF using $\kappa = (-2)$ was found often halt its operation for the reasons outlined in Section VII. For these reasons, we do not present the results of the EKF and the UKF here. As can be seen from Figs. 8(a), 8(b) and 8(c), the CKF marginally outperforms the CDKF. Moreover, the true RMSEs closely follow the RMSEs estimated by both filters; hence they are consistent. The overall results obtained for the CDKF and the CKF conform to our expectation because the considered scenario is a low-dimensional nonlinear-Gaussian filtering problem.

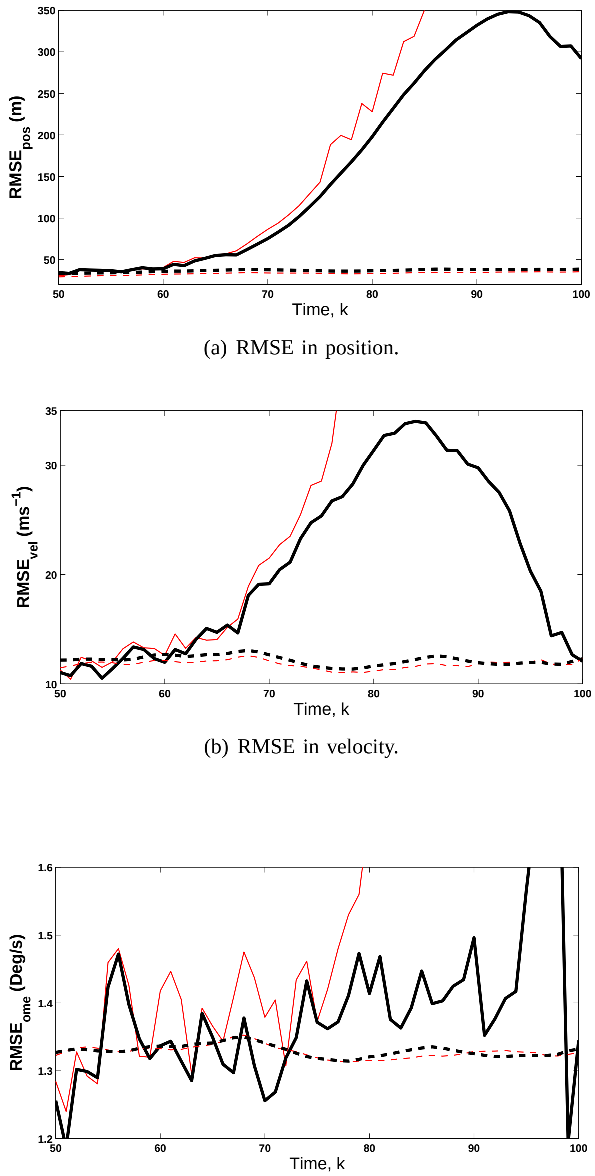
Scenario 2. We go on to extend the above problem to see how the estimation errors affect the CKF when the Gaussian nature of the problem is explicitly violated. To accomplish this, we deliberately make the measurement noise $\mathbf{w}_k$ to follow a Gaussian mixture of the form:

$$\mathbf{w}_k \sim 0.5\mathcal{N}(\mathbf{0}, R_1) + 0.5\mathcal{N}(\mathbf{0}, R_2),$$

where

$$R_1 = \begin{pmatrix} 100\text{m}^2 & 15\text{m mrad} \\ 15\text{m mrad} & 10\text{mrad}^2 \end{pmatrix}$$
$$R_2 = \begin{pmatrix} 5\text{m}^2 & 10\text{m mrad} \\ 10\text{m mrad} & 100\text{mrad}^2 \end{pmatrix}.$$

As shown in Fig. 9, the measurement noise density is now



(a) RMSE in position.

(b) RMSE in velocity.

(c) RMSE in turn rate.

Fig. 10. True RMSE (solid thin- CDKF, solid thick- CKF) Vs. filter-estimated RMSE (dashed thin- CDKF, dashed thick- CKF) for the second scenario.

highly asymmetric. The filters use an equivalent measurement noise covariance of the above Gaussian mixture model. Other data remain the same as before.

Not surprisingly, as can be seen from Figs. 10(a), 10(b) and 10(c), both filters exhibit divergence due to a mismatch between the filter-design assumption and the non-Gaussian nature of the problem. During the divergence period, the true RMSEs of both filters exceed their corresponding estimated RMSEs. Although there is little significant difference between covariance estimates produced by the two filters, it is encouraging to see that unlike the CDKF, the CKF improves the performance after a short period of divergence. To put it in another way, the CKF does not permit estimation errors to accumulate continuously, thereby avoiding an immediate 'blow-up' in this scenario.

IX. Conclusion

For a linear Gaussian dynamic system, the Kalman filter provides the optimal solution for the unknown state in the minimum mean squared-error sense. In general, however, real-world systems are plagued by nonlinearity, non-Gaussianity or both. In this context, it is difficult to obtain a closed-form solution for the state estimate, and therefore some approximations have to be made. Consequently, finding a more accurate nonlinear filtering solution has been the subject of intensive research since the celebrated work on the Kalman filter. Unfortunately, the presently known nonlinear filters, which are exemplified by the extended Kalman filter, the unscented Kalman filter, and the quadrature Kalman filter, experience divergence or the curse of dimensionality or both.

In this paper, we derived a more accurate nonlinear filter that could be applied to solve high-dimensional nonlinear filtering problems with minimal computational effort. The Bayesian filter solution in the Gaussian domain reduces to the problem of how to compute multi-dimensional integrals, whose integrands are all of the form *nonlinear function* × *Gaussian density*. Hence, we are emboldened to say that the cubature rule is the method of choice to solve this problem efficiently. Unfortunately, the cubature rule has been overlooked in the literature on nonlinear filters since the birth of the celebrated Kalman filter in 1960. We have derived a third-degree spherical-radial cubature rule to compute integrals numerically. We embedded the proposed cubature rule into the Bayesian filter to build a new filter, which we have named the cubature Kalman filter (CKF).

The cubature rule has the following desirable properties:

- The cubature rule is derivative-free. This useful property helps broaden the applicability of the CKF to situations where it is difficult or undesirable to compute Jacobians and Hessians. For example, consider a model configured by various model units having complicated nonlinearities; it may be inconvenient if we are required to calculate Jacobians and Hesssians for the resulting model. Additionally, a derivative-free computation allows us to write prepackaged, or 'canned' computer programs.
- The cubature rule entails $2n$ cubature points, where n is the state-vector dimension; hence, we are required to compute $2n$ functional evaluations at each update cycle. This suggests that the computational complexity scales linearly with the dimension n in terms of the functional evaluations whereas it grows cubically in terms of *flops*. Hence, the CKF eases the burden on the curse of dimensionality, but, by itself, it is not a complete remedy for the dimensionality issue.
- A third-degree cubature rule has a theoretical lower bound of $2n$ cubature points. We proved that a cubature rule of degree three is optimal when embedded in the Bayesian filter. Consequently, the CKF using the proposed spherical-radial cubature rule may be considered as an optimal approximation to the Bayesian filter that could be designed in a nonlinear setting, under the Gaussian assumption.

In a nutshell, the CKF is a new and improved algorithmic addition to the kit of tools for nonlinear filtering.

Appendix A
CKF Algorithm

Time Update

1) Assume at time k that the posterior density function $p(\mathbf{x}_{k-1}|D_{k-1}) = \mathcal{N}(\hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1})$ is known. Factorize

$$P_{k-1|k-1} = S_{k-1|k-1} S^T_{k-1|k-1}. \tag{38}$$

2) Evaluate the cubature points (i=1,2,...,m)

$$X_{i,k-1|k-1} = S_{k-1|k-1}\xi_i + \hat{\mathbf{x}}_{k-1|k-1}, \tag{39}$$

where $m = 2n_x$.

3) Evaluate the propagated cubature points (i=1,2,...,m)

$$X^*_{i,k|k-1} = \mathbf{f}(X_{i,k-1|k-1}, \mathbf{u}_{k-1}). \tag{40}$$

4) Estimate the predicted state

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X^*_{i,k|k-1}. \tag{41}$$

5) Estimate the predicted error covariance

$$\begin{aligned} P_{k|k-1} &= \frac{1}{m}\sum_{i=1}^{m} X^*_{i,k|k-1} X^{*T}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^T_{k|k-1} \\ &\quad + Q_{k-1}. \end{aligned} \tag{42}$$

Measurement Update

1) Factorize

$$P_{k|k-1} = S_{k|k-1} S^T_{k|k-1}. \tag{43}$$

2) Evaluate the cubature points (i=1,2,...,m)

$$X_{i,k|k-1} = S_{k|k-1}\xi_i + \hat{\mathbf{x}}_{k|k-1}. \tag{44}$$

3) Evaluate the propagated cubature points (i=1,2,...,m)

$$Z_{i,k|k-1} = \mathbf{h}(X_{i,k|k-1}, \mathbf{u}_k). \tag{45}$$

4) Estimate the predicted measurement

$$\hat{\mathbf{z}}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} Z_{i,k|k-1}. \tag{46}$$

5) Estimate the innovation covariance matrix

$$\begin{aligned} P_{zz,k|k-1} &= \frac{1}{m}\sum_{i=1}^{m} Z_{i,k|k-1} Z^T_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1} \\ &\quad + R_k. \end{aligned} \tag{47}$$

6) Estimate the cross-covariance matrix

$$P_{xz,k|k-1} = \sum_{i=1}^{m} \omega_i X_{i,k|k-1} Z^T_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}^T_{k|k-1}. \tag{48}$$

7) Estimate the Kalman gain

$$W_k = P_{xz,k|k-1}P^{-1}_{zz,k|k-1}. \quad (49)$$

8) Estimate the updated state

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + W_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}). \quad (50)$$

9) Estimate the corresponding error covariance

$$P_{k|k} = P_{k|k-1} - W_k P_{zz,k|k-1} W_k^T. \quad (51)$$

Appendix B
SCKF Algorithm

Below, we summarize the square-root cubature Kalman filter (SCKF) algorithm writing the steps explicitly when they differ from the CKF algorithm. We denote a general triangularization algorithm (e.g., the QR decomposition) as $S = \textbf{Tria}(A)$, where S is a lower triangular matrix. The matrices A and S are related as follows: Let R be an upper triangular matrix obtained from the QR decomposition on A^T; then $S = R^T$. We use the symbol / to represent the matrix right divide operator. When we perform the operation A/B, it applies the back substitution algorithm for an upper triangular matrix B and the forward substitution algorithm for a lower triangular matrix B:

Time Update

1) Skip the factorization (38) because the square-root of the error covariance, $S_{k-1|k-1}$, is available. Compute from (39)-(41).
2) Estimate the square-root factor of the predicted error covariance

$$S_{k|k-1} = \textbf{Tria}([\mathscr{X}^*_{k|k-1} \quad S_{Q,k-1}]). \quad (52)$$

where $S_{Q,k-1}$ denotes a square-root factor of Q_{k-1} such that $Q_{k-1} = S_{Q,k-1}S^T_{Q,k-1}$ and the weighted, centered (prior mean is subtracted off) matrix

$$\mathscr{X}^*_{k|k-1} = \frac{1}{\sqrt{m}}[X^*_{1,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \quad X^*_{2,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \dots X^*_{m,k|k-1} - \hat{\mathbf{x}}_{k|k-1}]. \quad (53)$$

Measurement Update

1) Skip the factorization (43) because the square-root of the error covariance, $S_{k|k-1}$, is available. Compute from (44)-(46).
2) Estimate the square-root of the innovation covariance matrix

$$S_{zz,k|k-1} = \textbf{Tria}([\mathscr{Z}_{k|k-1} \quad S_{R,k}]) \quad (54)$$

where $S_{R,k}$ denotes a square-root factor of R_k such that $R_k = S_{R,k}S^T_{R,k}$ and the weighted, centered matrix

$$\mathscr{Z}_{k|k-1} = \frac{1}{\sqrt{m}}[Z_{1,k|k-1} - \hat{\mathbf{z}}_{k|k-1} \quad Z_{2,k|k-1} - \hat{\mathbf{z}}_{k|k-1} \dots Z_{m,k|k-1} - \hat{\mathbf{z}}_{k|k-1}]. \quad (55)$$

3) Estimate the cross-covariance matrix

$$P_{xz,k|k-1} = \mathscr{X}_{k|k-1}\mathscr{Z}^T_{k|k-1} \quad (56)$$

where the weighted, centered matrix

$$\mathscr{X}_{k|k-1} = \frac{1}{\sqrt{m}}[X_{1,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \quad X_{2,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \dots X_{m,k|k-1} - \hat{\mathbf{x}}_{k|k-1}]. \quad (57)$$

4) Estimate the Kalman gain

$$W_k = (P_{xz,k|k-1}/S^T_{zz,k|k-1})/S_{zz,k|k-1}. \quad (58)$$

5) Estimate the updated state $\hat{\mathbf{x}}_{k|k}$ as in (50).
6) Estimate the square-root factor of the corresponding error covariance

$$S_{k|k} = \textbf{Tria}([\mathscr{X}_{k|k-1} - W_k\mathscr{Z}_{k|k-1} \quad W_kS_{R,k}]). \quad (59)$$

Here we derive the square-root error covariance as follows: Substituting (49) into (51) yields

$$P_{k|k} = P_{k|k-1} - P_{xz,k|k-1}K^T. \quad (60)$$

Some matrix manipulations in (49) lead to

$$KP^T_{xz,k|k-1} = KP_{zz,k|k-1}K^T. \quad (61)$$

Adding (60) and (61) together yields

$$P_{k|k} = P_{k|k-1} - P_{xz,k|k-1}K^T + KP_{zz,k|k-1}K^T - KP^T_{xz,k|k-1}. \quad (62)$$

Using the fact that $P_{k|k-1} = \mathscr{X}_{k|k-1}\mathscr{X}^T_{k|k-1}$ and substituting (54), and (56) into (62) appropriately, we have

$$\begin{aligned} P_{k|k} &= \mathscr{X}_{k|k-1}\mathscr{X}^T_{k|k-1} - \mathscr{X}_{k|k-1}\mathscr{Z}^T_{k|k-1}K^T \\ &\quad + K(\mathscr{Z}_{k|k-1}\mathscr{Z}^T_{k|k-1} + S_{R,k}S^T_{R,k})K^T \\ &\quad - K\mathscr{Z}_{k|k-1}\mathscr{X}^T_{k|k-1} \\ &= [\mathscr{X}_{k|k-1} - K\mathscr{Z}_{k|k-1} \quad KS_{R,k}] \\ &\quad \times [\mathscr{X}_{k|k-1} - K\mathscr{Z}_{k|k-1} \quad KS_{R,k}]^T. \end{aligned} \quad (63)$$

$P_{k|k}$ in (63) is the Joseph covariance in disguise. Hence, the matrix triangularization of (63) leads to (59).

Acknowledgement

The authors would like to thank the associate editor and the anonymous reviewers for many constructive comments that helped them to clarify the presentation.

References

[1] Y. C. Ho and R. C. K. Lee, “A Bayesian approach to problems in stochastic estimation and control,” *IEEE Trans. Automatic Cont.*, vol. AC-9, pp. 333-339, Oct. 1964.

[2] V. Peterka, “Bayesian approach to system identification,” in *Trends and progress in system identification*, P. Eykhoff, ed., pp. 239-304, Pergamon Press, Oxford, 1981.

[3] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *ASME J. Basic Eng.*, vol. 82, pp. 34-45, March 1960.

[4] J. S. Liu, *Monte Carlo strategies in scientific computing*, Springer, 2001.

[5] V. E. Benes, “Exact finite-dimensional filters with certain diffusion nonlinear drift,” *Stochastics*, vol. 5, pp. 65-92, 1981.

[6] H. J. Kushner, “Approximations to optimal nonlinear filters,” *IEEE Trans. Automatic Cont.*, AC-12(5), pp. 546-556, Oct. 1967.

[7] B. D. O. Anderson, and J. B. Moore, *Optimal filtering*, Eaglewood Cliffs, NJ:Prentice-Hall, 1979.

[8] T. S. Schei, “A Finite difference method for linearizing in nonlinear estimation algorithms,” *Automatica*, vol. 33, no. 11, pp. 2051-2058, 1997.

[9] M. Nørgaard, N. K. Poulsen, and O. Ravn, “New developments in state estimation of nonlinear systems, ” *Automatica*, vol. 36, pp.1627-1638, 2000.

[10] S. J. Julier, J.K. Ulhmann, and H. F. Durrant-Whyte, “A new method for nonlinear transformation of means and covariances in filters and estimators,” *IEEE Trans. Automatic Cont.*, vol. 45, pp. 472-482, March 2000.

[11] K. Ito, and K. Xiong, “Gaussian filters for nonlinear filtering problems,” *IEEE Trans. Automat. Control*, vol. 45, no. 5, pp. 910-927, May 2000.

[12] 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.

[13] M. Simandl, J. Kralovec and T. Söderstörm, “Advanced point-mass method for nonlinear state estimation,” *Automatica*, vol. 42, no. 7, pp. 1133-1145, 2006.

[14] D. L. Alspach, and H. W. Sorenson, “Nonlinear Baysian estimation using Gaussian sum approximations,” *IEEE Trans. Automatic Cont.*, vol. 17, no. 4, pp. 439-448, Aug. 1972.

[15] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation, ” *IEE Proc.-F*, vol. 140, pp. 107-113, April 1993.

[16] O. Cappe, S. J. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential Monte carlo,” *Proc. IEEE* vol. 95, no. 5, May 2007.

[17] R. E. Bellman, *Adaptive control processes*, NJ: Princeton Univ. Press, 1961.

[18] A. H. Stroud, *Approximate calculation of multiple integrals*, NJ: Prentice Hall, 1971.

[19] C. Stone, *A course in probability and statistics*, Duxbury Press, 1996.

[20] T. Kailath, “A view of three decades of linear filterng theory,” *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 146-181, Mar. 1974.

[21] A. H. Stroud, *Gaussian quadrature formulas*, NJ: Prentice Hall, 1966.

[22] J. Stoer, and R. Bulirsch, *Introduction to Numerical Analaysis*, 3rd ed., Springer, 2002.

[23] R. Cools, “Advances in multidimensional integration,” *J. of Comput. & Applied Math.*, vol. 149, no. 1, pp. 1-12, Dec. 2002.

[24] H. Niederreiter, “Quasi-Monte Carlo methods and pseudo-random numbers” *Bull. Amer. Math. Soc.*, vol. 84, pp. 957 − 1041, 1978.

[25] P. Lecuyer, and C. Lemieux, *Recent advances in randomized quasi-Monte Carlo methods*, Ch. 20 in *Modeling uncertainty: An examination of stochastic theory, methods and applications*, MA: Kluwer Academic, 2002.

[26] I. H. Sloan, and S. Joe, *Lattice methods for multiple integration*, Oxford: Oxford Univ. Press, 1994.

[27] S. A. Smolyak, “Quadrature and interpolation formulas for tensor products of certain classes of functions,” *Soviet. Math. Dokl. 4*, pp. 240-243, 1963.

[28] A. Genz, and B. D. Keister, “Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight,” *J. Comput. Appl. Math.*, vol. 71, no. 2, pp. 299-309, 1996.

[29] T. Gerstner, and M. Griebel, “Numerical integration using sparse grids” *Numerical Algorithms*, vol. 18, no. 4, pp. 209 232, 1998.

[30] F. Daum, “Nonlinear filters: Beyond the Kalman filter,” *Aerospace and Electronic Sys. Magazine*, vol. 20, no. 8, pp. 57-69, Aug. 2005.

[31] R. Cools, “Constructing cubature formulas: the science behind the art,” *Acta Numerica 6*, pp. 1 54, 1997.

[32] S.L. Sobolev, “Formulas of mechanical cubature on the surface of a sphere,” *Sibirsk. Math.* vol. Z.3, pp. 769 796, 1962 (Russian).

[33] W. H. Press, and S. A. Teukolsky, “Orthogonal polynomials and Gaussian quadrature with nonclassical weighting functions,” *Computers in Physics*, pp. 423-426, july/aug. 1990.

[34] G. M. Philips, “A survey of one-dimensional and multi-dimensional numerical integration,” *Comp. Physics Comm.*, vol. 20, pp. 17-27, 1980.

[35] P. Rabinowitz, and N. Richter, “Perfectly symmetric two-dimensional integration formulas with minimal numbers of points,” *Math. of Computation*, vol. 23, no. 108, pp. 765-779, Oct., 1969.

[36] M. Grewal, and A. Andrews, *Kalman filtering: Theory and practice using Matlab*, 2nd ed., NY:Wiley, 2001.

[37] P. Kaminski, A. Bryson, and S. Schmidt, “Discrete square root filtering: A survey of current techniques,” *IEEE Trans. Automat. control*, vol. ac-16, Dec. 1971.

[38] I. Arasaratnam, and S. Haykin, “Square-root quadrature Kalman filtering”, *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2589-2593, June 2008.

[39] S. J. Julier, and J. K. Ulhmann, “Unscented filtering and nonlinear estimation”, *Proc. IEEE*, vol. 92, no. 3, March 2004.

[40] R. van der Merwe, and E. Wan, “The square-root unscented Kalman filter for state and parameter estimation,” *Proc. Int’l Conf. Acous. Speech, and Signal Process. (ICASSP01)*, vol. 6, pp. 3461-3464, 2001.

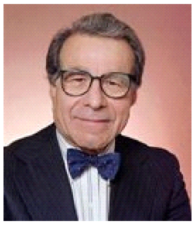
[41] S. J. Julier, and J. K. Uhlmann, “The scaled unscented transformation,” *Proc. IEEE American Control Conf.*, pp. 4555 4559, USA, May 2002.

[42] Y. Bar Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*, NY: Wiley & Sons, 2001.

Ienkaran Arasaratnam received the B.Sc. degree (first-class honors) in the Department of Electronics and Telecommunication Engineering from the University of Moratuwa, Sri Lanka in 2003 and the M.A.Sc. degree in the Department of Electrical and Computer Engineering from McMaster University, Hamilton, ON, Canada in 2006.

Since Jan. 2006, he has been pursuing his Ph.D. degree at McMaster University. His main research interests include nonlinear estimation, control and machine learning.

Mr. Arasaratnam received the Mahapola Merit Scholarship during his undergraduate studies. He is the recipient of the Outstanding Thesis Research Award at M.A.Sc. Level in 2006 and the Ontario Graduate Scholarship in Science and Technology in 2008.



Simon Haykin (Fellow, IEEE) received the B.Sc. (first-class honors), Ph.D., and D.Sc. degrees, all in Electrical Engineering from the University of Birmingham, England.

Currently, he is the University Professor at McMaster University, Hamilton, ON, Canada. He is a pioneer in adaptive signal-processing with emphasis on applications in radar and communications, an area of research which has occupied much of his professional life. In the mid 1980s, he shifted the thrust of his research effort in the direction of Neural Computation, which was re-emerging at that time. All along, he had the vision of revisiting the fields of radar and communications from a brand new perspective. That vision became a reality in the early years of this century with the publication of two seminal journal papers: (i) Cognitive Radio: Brain-empowered Wireless communications, IEEE J. Selected Areas in Communications, Feb. 2005 (ii) Cognitive Radar: A Way of the Future, IEEE J. Signal Processing, Feb. 2006. Cognitive Radio and Cognitive Radar are two important parts of a much wider and multidisciplinary subject: Cognitive Dynamic Systems, research into which has become his passion.

Prof. Haykin is a Fellow of the Royal Society of Canada. He is the recipient of the Henry Booker Medal of 2002, the Honorary Degree of Doctor of Technical Sciences from ETH Zentrum, Zurich, Switzerland, in 1999, and many other medals and prizes.