IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 19, 2021, accepted April 7, 2021, date of publication April 14, 2021, date of current version April 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073289

Extended Kalman Filter Using Orthogonal Polynomials

KUNDAN KUMAR[1], (Graduate Student Member, IEEE), SHOVAN BHAUMIK[1], AND PARESH DATE[2]

[1]Department of Electrical Engineering, Indian Institute of Technology Patna, Patna 801106, India
[2]Department of Mathematics, College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, U.K.

Corresponding author: Kundan Kumar (kundan.pee16@iitp.ac.in)

ABSTRACT This paper reports a new extended Kalman filter where the underlying nonlinear functions are linearized using a Gaussian orthogonal basis of a weighted $\mathcal{L}_2$ space. As we are interested in computing the states' mean and covariance with respect to Gaussian measure, it would be better to use a linearization, that is optimal with respect to the same measure. The resulting first-order polynomial coefficients are approximately calculated by evaluating the integrals using (i) third-order Taylor series expansion (ii) cubature rule of integration. Compared to direct integration-based filters, the proposed filter is far less susceptible to the accumulation of round-off errors leading to loss of positive definiteness. The proposed algorithms are applied to four nonlinear state estimation problems. We show that our proposed filter consistently outperforms the traditional extended Kalman filter and achieves a competitive accuracy to an integration-based square root filter, at a significantly reduced computing cost.

INDEX TERMS State estimation, filtering, Kalman filter, functional approximation, Taylor series, numerical analysis, orthogonal polynomial, nonlinear filter, target tracking, computational efficiency.

I. INTRODUCTION

A. BAYESIAN FILTERING

We consider the following state space model of a dynamic system in discrete time:

$$\mathcal{X}_{k+1} = \phi(\mathcal{X}_k) + \eta_k, \tag{1}$$

and

$$\mathcal{Y}_{k+1} = \gamma(\mathcal{X}_{k+1}) + \nu_{k+1}, \tag{2}$$

where $\mathcal{X}_k \in \mathbb{R}^{n_x}$ is the state of the dynamic system at a time instant k, $\mathcal{Y}_k \in \mathbb{R}^{n_y}$ is the measurement, $\phi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $\gamma : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ are known nonlinear functions. The process noise, η_k, and the measurement noise, ν_{k+1} are white, Gaussian and uncorrelated to each other with zero mean and covariance Q_k and R_{k+1}, respectively.

In Bayesian filtering, the hidden state vector $\mathcal{X}_{k+1}$ needs to be estimated by using measurement up to time $k+1$. The pdf $p(\mathcal{X}_{k+1}|\mathcal{Y}_{1:k+1})$ is constructed recursively in two steps: (i) prediction step (ii) update step. In the prediction step, from the knowledge of $p(\mathcal{X}_k|\mathcal{Y}_{1:k})$, we construct the pdf $p(\mathcal{X}_{k+1}|\mathcal{Y}_{1:k})$ using the Chapman-Kolmogorov equation:

$$p(\mathcal{X}_{k+1}|\mathcal{Y}_{1:k}) = \int p(\mathcal{X}_{k+1}|\mathcal{X}_k)p(\mathcal{X}_k|\mathcal{Y}_{1:k})d\mathcal{X}_k. \tag{3}$$

In the update step, the posterior density function of the state $p(\mathcal{X}_{k+1}|\mathcal{Y}_{1:k+1})$ is computed using Bayes' theorem:

$$p(\mathcal{X}_{k+1}|\mathcal{Y}_{1:k+1}) \propto p(\mathcal{Y}_{k+1}|\mathcal{X}_{k+1})p(\mathcal{X}_{k+1}|\mathcal{Y}_{1:k}). \tag{4}$$

For a linear Gaussian system, closed-form solution of the Eqs. (3) – (4) is available and it is known as the Kalman filter (KF) [1], [2]. The KF is the optimal conditional mean estimator and provides the minimum mean squared error (MMSE). For a nonlinear system, no such closed optimal solution is available in general and many heuristic approaches exist to obtain an approximate solution; in particular, the first two moments (the mean and the covariance matrix) of the states.

B. BRIEF LITERATURE SURVEY

The extended Kalman filter (EKF) [2], [3] uses local linearization using Taylor series expansion and then uses the KF recursions for the resulting system. The EKF has some attractive properties such as local stability, ease of implementation, and modest computational resources compared

The associate editor coordinating the review of this manuscript and approving it for publication was Gerard-Andre Capolino.

to some of the alternative filtering heuristics. By local stability, we mean that the covariance matrix of the EKF can be shown to remain bounded under appropriate assumptions (mild nonlinearity, bounded noise covariance, and initial estimated errors *etc*.) [4, pp. 46–53], [5]. The EKF became very popular in the 1960s and is still being applied to many real-life problems. However, it suffers from the disadvantage of track divergence when the system is highly nonlinear.

To improve the accuracy of estimation without a significant increase in computational cost, a set of filters collectively known as deterministic sample point filters are proposed. The unscented Kalman filter (UKF) [6], [7], the Gauss-Hermite filter (GHF) [8], the cubature Kalman filter (CKF) [9] and its variance [10]–[12] are examples of such filters. In these filters, like the EKF, the prior and the posterior pdfs are assumed to be Gaussian, and the mean and the covariance are calculated using specifically chosen sample points and weights. While linearity and Gaussianity are assumed in the computation of conditional moments, no explicit linearization is needed (unlike the EKF). Reference [13] provides an extensive review of filtering using deterministic sample points. The computational cost of such filters is somewhat higher than the EKF, but it is still substantially lower than Monte Carlo based filters.

In another development, the posterior and prior pdf of states are approximated with appropriately chosen points in the state space (known as particles) and the associated probability weights [14]. Such a method of estimation is known as particle filter (PF) [14]. The computational cost of a particle filter is very high and it shows the curse of dimensionality problem. To estimate the states of a very large dimensional system, the ensemble Kalman filter (EnKF) was proposed in [15]. In the EnKF, we generate a set of ensembles using Monte Carlo sampling, and propagate it through the state and measurement equations, and subsequently update it with Kalman filtering scheme. In another approach, Dunik *et. al.* proposed stochastic integration filter (SIF) [16], [17] based on the stochastic integration rules (SIRs). It combines the Monte Carlo and sigma point method into a single filtering algorithm. The computation complexity of the SIF is substantially lower than the sequential Monte Carlo filters.

Staying within the explicit linearization framework (mainly due to small execution time), a few heuristic fixes exist for addressing track divergence and poor performance issues of the EKF under significant nonlinearities, including iterative improvement of the local linear estimate; see [13] and references therein for more details. However, Taylor series expansion still remains the key to approximate a nonlinear function in these methods.

C. CONTRIBUTION

In this paper, we suggest to use a different linearization technique which is a first-order polynomial approximation with a Gaussian measure as a weighting function. Essentially, this uses the first two polynomials from an orthogonal basis of a weighted $\mathcal{L}_2$ space. Most functions that appear in nonlinear filtering problems belong to this space. The coefficients of the first order polynomial are approximately calculated by evaluating the integrals using (i) Taylor series approximation of the function (ii) numerical evaluation of the Gaussian integral using a weighted sum of an appropriate set of deterministic sample points. The first approach assumes that the process and the measurement functions are thrice differentiable, and a third-order Taylor series approximation is used to calculate the coefficients. We call this Taylor series orthogonal polynomial-based extended Kalman filter as TO-EKF. For the second approach, we use spherical radial cubature rule [9] for evaluating integrals and hence computing the coefficients of linearized function. This second approach is referred to as CO-EKF. Once we have a linear approximation, the remaining computation during a single recursion step is identical to the KF. Note that one can also use other quadrature integration methods such as Gauss-Hermite quadrature [8] for integration.

We apply the CO-EKF and the TO-EKF to four nonlinear state estimation problems. The performances of these filtering techniques are compared to the EKF and the CKF (or square root CKF when we need to address the Cholesky decomposition error [18], [19]) in terms of root mean square error (RMSE), the track loss percentage, the average normalized (state) estimation error squared (NEES), the bias norm, and the relative run time.

D. CLAIM

i. It is to be noted that the Taylor series approximation is a purely local approximation and is not optimal in any norm; its accuracy tends to degrade rapidly away from the nominal point. The proposed filters are expected to perform better than the Taylor series based EKF since the chosen basis is optimal with respect to Gaussian measure [20], [21], in a certain, formal sense and we are using linearization principally to integrate with respect to Gaussian measure (to compute the first two moments). The simulation results support our claim and we see that the proposed TO-EKF and CO-EKF provide better estimation accuracy than the EKF. Further, the CO-EKF shows a similar estimation performance compared to the SRCKF at a far lower computational cost.

ii. Compared to other deterministic sample point-based filters such as the CKF and the GHF, the proposed TO-EKF or CO-EKF offer significant computational stability. Any potential loss of positive definiteness or symmetry due to numerical errors can be addressed far more easily in the EKF framework, *e.g.* using the Joseph form. In the CKF, the roundoff errors are usually dealt with using a square root implementation, which is computationally far more expensive [9], [18]. Later in section VII, we justify the above claim in terms of floating-point operations.

II. LINEAR APPROXIMATION USING ORTHOGONAL POLYNOMIALS

Our objective is to approximately evaluate an arbitrary nonlinear function $\phi(\mathcal{X})$, $\phi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_m}$, where $\mathcal{X} \in \mathbb{R}^{n_x}$ is a Gaussian random variable (r.v.) with mean $\hat{\mathcal{X}}$ and covariance $P \in \mathbb{R}^{n_x \times n_x}$. We first transform $\mathcal{X}$ into another r.v. x with standard normal distribution:

$$\mathcal{X} = \hat{\mathcal{X}} + Sx, \tag{5}$$

where S is a square root matrix of P *i.e.* $P = SS^T$. Now, we would like to approximate the nonlinear function $\phi(\hat{\mathcal{X}} + Sx)$, using an appropriate basis. We will start with the definition for scalar functions $n_x = 1$ and then extend it suitably to vector-valued functions of vector-valued variables. For $n_x = 1$, consider an inner product space of functions which are square integrable with respect to standard normal weighting function:

$$S_{\mathcal{N}} = \left\{ f : \int_{-\infty}^{\infty} \mathcal{N}(x; 0, 1) f(x)^2 \, dx < \infty \right\},$$

where $\mathcal{N}(x; 0, 1) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$. It is known that Hermite polynomials $h_i(x)$, $i = 0, 1, 2, \cdots$ form an orthonormal basis for this space, where

$$h_i(x) = (-1)^i \mathcal{N}^{-1}(x) \frac{d^i \mathcal{N}(x)}{dx^i}, \; h_0(x) = 1,$$

and $\mathcal{N}(x) = \mathcal{N}(x; 0, 1)$. The first three polynomials are given by $h_0(x) = 1, h_1(x) = x$ and $h_2(x) = x^2 - 1$. It is easy to verify that the polynomials satisfy orthogonality condition [22] *i.e.*

$$\int_{-\infty}^{\infty} h_i(x) h_j(x) \mathcal{N}(x; 0, 1) dx = \begin{cases} i! & \text{for } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Any nonlinear function $f(x) \in S_{\mathcal{N}}$, can be approximated as

$$f(x) = \sum_{i=0}^{N} a_i h_i(x), \tag{7}$$

where $N \in \mathbb{N}$ is the order of approximation. The coefficients a_i are expressed as

$$a_i = \int_{-\infty}^{\infty} f(x) h_i(x) \mathcal{N}(x; 0, 1) dx. \tag{8}$$

For recursive relation between $h_i(x)$ and other important properties readers are referred to [23, p. 775]. The concept described above is extended for approximating a nonlinear function in multidimensional real space *i.e.* for $n_x > 1$. Let $f = \begin{bmatrix} f_1 \; f_2 \; \cdots \; f_{n_p} \end{bmatrix}^T$ be a vector in $\mathbb{R}^{n_p}$ be a vector valued function of vector valued variable $x = \begin{bmatrix} x_1 \; x_2 \; \cdots \; x_{n_x} \end{bmatrix}^T$ and let $n = \begin{bmatrix} n_1 \; n_2 \; \cdots \; n_{n_x} \end{bmatrix}^T$ be a vector of polynomial orders in $\mathbb{N}^{n_x}$. Multivariate Hermite polynomials of order $n' = n_1 + n_2 + \cdots + n_{n_x}$, is usually defined by a product of scalar Hermite polynomials:

$$H_n(x) = \frac{1}{\sqrt{n!}} \Pi_{j=1}^{n_x} h_{n_j}(x_j), \tag{9}$$

where $n! = \Pi_{l=1}^{n_x} n_l!$. These polynomials also form an orthonormal basis for the inner product space of square integrable functions which map $\mathbb{R}^{n_x}$ to $\mathbb{R}$:

$$\int_{-\infty}^{\infty} \mathcal{N}(x; 0, I) H_m(x) H_n(x) dx = \begin{cases} 1 & \text{for } n = m, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $\mathcal{N}(x; 0, I) = \frac{1}{(2\pi)^{\frac{n_x}{2}}} \exp(-\frac{x^T x}{2})$. As an example, if we consider $n_x = 2$, zeroth order Hermite polynomial will be $H_{00} = 1$, first order Hermite polynomials will be $H_{10} = x_1$ and $H_{01} = x_2$, and second order Hermite polynomials will be $H_{11} = x_1 x_2$, $H_{20} = \frac{1}{\sqrt{2}}(x_1^2 - 1)$, and $H_{02} = \frac{1}{\sqrt{2}}(x_2^2 - 1)$. Higher order multivariate Hermite polynomials have more involved expressions. Assuming that each j-th element of f which we write as f_j, belongs to the space

$$S'_{\mathcal{N}} = \left\{ f : \int_{-\infty}^{\infty} \mathcal{N}(x; 0, I) f(x)^2 \, dx < \infty \right\},$$

can be expressed $f_j(x)$ in terms of this orthonormal basis as

$$f_j(x) = \sum_{i \in \mathbb{N}^{n_x}} a_i^j H_i(x), \tag{11}$$

where

$$a_i^j = \int_{-\infty}^{\infty} f_j(x) H_i(x) \mathcal{N}(x; 0, I) dx, \tag{12}$$

and the last integral is over n_x dimensions for any choice of vector subscript i. It is worth mentioning here that, to evaluate the above integral for an arbitrary nonlinear function, we can follow one of two approaches: (i) use a Taylor series approximation of $f_j(x)$, which leads to the integration of a polynomial in terms of standard Gaussian measure; (ii) use numerical integration using an appropriate set of deterministic points. In this paper, we describe both these methods in sections IV and V, respectively. As we are only interested in a linear approximation, we shall confine ourselves with up to first-order Hermite polynomial *i.e.* $n' = 0$ and $n' = 1$. With this truncation and for $n_x = 2$, the j-th element of the function is approximated as

$$\begin{aligned} f_j(x) &\approx \sum_{i=\{00,01,10\}} a_i^j H_i(x) \\ &= a_{00}^j H_{00}(x) + a_{01}^j H_{01}(x) + a_{10}^j H_{10}(x). \end{aligned} \tag{13}$$

The function $f(x)$ which is a vector-valued function can be written as

$$f(x) \approx \mathbf{a_0} + \mathbf{a_1} \mathbf{H_1}(x), \tag{14}$$

where $\mathbf{H_1}(x) = [H_{10\cdots0} \; H_{01\cdots0} \; \cdots \; H_{00\cdots1}]^T_{(n_x \times 1)}$, is a column vector of first order Hermite polynomials, $\mathbf{a_0}$ and $\mathbf{a_1}$ are matrices with dimension $(n_p \times 1)$ and $(n_p \times n_x)$, respectively. We choose the Hermite polynomial because the weighting function of the Hermite polynomial is same as the probability density function of the Gaussian random variables. More details on multivariate Hermite polynomials can be found in [22], [24] and references therein.

Adding a time subscript and changing notation to be consistent with standard notation used for state-space models, we can write a first-order approximation of $\phi(\mathcal{X})$ and $\gamma(\mathcal{X})$ in the original problem as

$$\phi(\mathcal{X}_k) = \phi(\hat{\mathcal{X}}_{k|k} + S_{k|k}x_k) \approx A_k x_k + B_k, \tag{15}$$

and

$$\gamma(\mathcal{X}_{k+1}) = \gamma(\hat{\mathcal{X}}_{k+1|k} + S_{k+1|k}x_{k+1}) \approx C_k x_{k+1} + D_k. \tag{16}$$

The matrices A_k, C_k and the vectors B_k, D_k can be evaluated using Eq. (12). Note that this 'linearization' of nonlinear functions is fundamentally different from the gradient-based linearization used in the EKF.

In the literature, there are a few papers which either use the Gauss-Hermite series or include 'polynomial filtering' in the title. Reference [25] proposed a filtering algorithm (named the Fourier Hermite Kalman filter (FHKF)), based on the finite truncation of the Fourier-Hermite series. In the FHKF, the expected value of a nonlinear function has to be expressed in closed form, which is always not possible for any arbitrary nonlinear function [26, p. 80]. An algorithm based on the second-order polynomial chaos approximation is proposed in [27] and it is named polynomial chaos Kalman filter (PCKF). It uses a set of support points and Hermite polynomial and is much closer, as an algorithm, to the Gauss Hermite filter (GHF) than our filter. The algorithm presented in [28] uses Carleman approximation and its run time is very high. [29] developed a filtering method for a single-dimensional system that intrinsically presents polynomial functions by exploiting the Taylor series to derive second-order statistics. None of the approaches look at optimal linear approximation in weighted $\mathcal{L}_2$ space, in the sense of the work presented here. Reader should also be aware that the Gauss-Hermite filter and Hermite polynomial approximation used here are entirely different approaches even though the word 'Hermite' is common in both.

III. EKF WITH ORTHOGONAL POLYNOMIALS

The integral mentioned in Eq. (12) has no closed-form in general and we need to solve it numerically. For the time being, let us assume that the integral is evaluated by some means in each case for a first order polynomial approximation of $\phi(\mathcal{X}_k)$ and $\gamma(\mathcal{X}_k)$, and we have calculated the coefficients $A_k \in \mathbb{R}^{n_x \times n_x}$, $B_k \in \mathbb{R}^{n_x}$, $C_k \in \mathbb{R}^{n_y \times n_x}$ and $D_k \in \mathbb{R}^{n_y}$. From Eqs. (15) – (16), the process and the measurement equation mentioned in Eqs. (1) – (2) become

$$\mathcal{X}_{k+1} \approx A_k x_k + B_k + \eta_k, \tag{17}$$
$$\mathcal{Y}_{k+1} \approx C_k x_{k+1} + D_k + \nu_{k+1}. \tag{18}$$

It is worth emphasizing again that x_k follows the standard normal distribution. With the above linearization, expressions of the prior and the posterior estimate and error covariance can be calculated as follows:

PREDICTION STEP

The prior estimate of the state can be given as

$$\begin{aligned}\hat{\mathcal{X}}_{k+1|k} &= E[\mathcal{X}_{k+1}|\mathcal{Y}_k] \\ &= \int_{-\infty}^{\infty}(A_k x_k + B_k + \eta_k)\mathcal{N}(x_k; 0, I)dx_k \\ &= B_k. \end{aligned} \tag{19}$$

The prior error covariance can be computed as

$$\begin{aligned}P_{k+1|k} = \int_{-\infty}^{\infty}&(A_k x_k + B_k - \hat{\mathcal{X}}_{k+1|k})(A_k x_k + B_k \\ &- \hat{\mathcal{X}}_{k+1|k})^T \mathcal{N}(x_k; 0, I)dx_k + Q_k,\end{aligned}$$

or,

$$\begin{aligned}P_{k+1|k} &= \int_{-\infty}^{\infty}(A_k x_k x_k^T A_k^T)\mathcal{N}(x_k; 0, I)dx_k + Q_k \\ &= A_k A_k^T + Q_k. \end{aligned} \tag{20}$$

UPDATE STEP

In this step, after receiving a measurement $\mathcal{Y}_{k+1}$, the posterior density function of the state $\mathcal{X}_{k+1}$ is computed by Bayes' rule as [2]

$$\begin{aligned}\hat{\mathcal{Y}}_{k+1|k} &= E[\mathcal{Y}_{k+1}|\mathcal{Y}_k] \\ &= \int_{-\infty}^{\infty}(C_k x_{k+1} + D_k + \nu_{k+1})\mathcal{N}(x_{k+1}; 0, I)dx_{k+1},\end{aligned}$$

or,

$$\hat{\mathcal{Y}}_{k+1|k} = D_k. \tag{21}$$

The measurement error covariance can be computed as

$$\begin{aligned}P^{\mathcal{Y}\mathcal{Y}}_{k+1|k} = \int_{-\infty}^{\infty}&(C_k x_{k+1} + D_k - \hat{\mathcal{Y}}_{k+1|k}) \\ &\times (C_k x_{k+1} + D_k - \hat{\mathcal{Y}}_{k+1|k})^T \\ &\mathcal{N}(x_{k+1}; 0, I)dx_{k+1} + R_{k+1}\end{aligned} \tag{22}$$

$$\begin{aligned}&= \int_{-\infty}^{\infty}(C_k x_{k+1} x_{k+1}^T C_k^T)\mathcal{N}(x_{k+1}; 0, I)dx_{k+1} + R_{k+1} \\ &= C_k C_k^T + R_{k+1}. \end{aligned} \tag{23}$$

The cross-covariance between state and measurement can be expressed as

$$\begin{aligned}P^{\mathcal{X}\mathcal{Y}}_{k+1|k} &= \int_{-\infty}^{\infty} S_{k+1|k}x_{k+1}(C_k x_{k+1} + D_k - \hat{\mathcal{Y}}_{k+1|k})^T \\ &\qquad \mathcal{N}(x_{k+1}; 0, I)dx_{k+1} \\ &= \int_{-\infty}^{\infty} S_{k+1|k}x_{k+1}x_{k+1}^T C_k^T \mathcal{N}(x_{k+1}; 0, I)dx_{k+1} \\ &= S_{k+1|k}C_k^T. \end{aligned} \tag{24}$$

The reader may note that S can be calculated from P using Cholesky decomposition. The posterior mean and covariance can be calculated using the following equations:

$$\hat{\mathcal{X}}_{k+1|k+1} = \hat{\mathcal{X}}_{k+1|k} + K_{k+1}(\mathcal{Y}_{k+1} - \hat{\mathcal{Y}}_{k+1|k}), \tag{25}$$
$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1}P^{\mathcal{Y}\mathcal{Y}}_{k+1|k}K_{k+1}^T, \tag{26}$$

where K_{k+1} is the Kalman gain which is calculated as

$$K_{k+1} = P^{\mathcal{XY}}_{k+1|k}(P^{\mathcal{YY}}_{k+1|k})^{-1}. \tag{27}$$

As we mentioned earlier, there is no exact solution of the integral mentioned in Eq. (12), we adopt two different approaches to evaluate this integral (and hence to compute the coefficients A_k, B_k, C_k and D_k), which are described in the next two sections. The first approach assumes that the $\phi(\hat{\mathcal{X}}_{k|k}+S_{k|k}x_k)$ and $\gamma(\hat{\mathcal{X}}_{k+1|k}+S_{k+1|k}x_{k+1})$ are sufficiently smooth and a third-order Taylor series approximation of these functions is used to carry out the necessary integration. In another approach, a few deterministic sample points and their associated weights are generated. A weighted sum of these deterministic sample points provides the value of the integral. In this approach, the differentiability of the function is not required.

IV. CALCULATION OF A_k, B_k, C_k AND D_k USING THE TAYLOR SERIES APPROXIMATION

For simplicity of exposition, we will first use the notation $\tilde{\phi}(x_k)$ and $\tilde{\gamma}(x_{k+1})$ in place of $\phi(\hat{\mathcal{X}}_{k|k}+S_{k|k}x_k)$ and $\gamma(\hat{\mathcal{X}}_{k+1|k}+S_{k+1|k}x_{k+1})$, respectively, and derive results in terms of partial derivatives of $\tilde{\phi}$, $\tilde{\gamma}$. Then we will evaluate these partial derivatives in terms of derivatives of ϕ, γ and hence derive the necessary expressions for A_k, B_k, C_k and D_k.

Proposition 1: If $\tilde{\phi}(x_k)$ is thrice differentiable at 0, the expressions for A_k and B_k in the third order Taylor series approximation for Eq. (15) *can be written as*

$$A_k = \sum_{i=1}^{n_x} \frac{\partial\tilde{\phi}(x_k)}{\partial x_{i,k}}\bigg|_{x_k=0} e_i^T + \frac{1}{2}\sum_{i=1}^{n_x}\sum_{j=1}^{n_x} \frac{\partial^3\tilde{\phi}(x_k)}{\partial x_{i,k}^2\,\partial x_{j,k}}\bigg|_{x_k=0} e_j^T,$$
$$B_k = \tilde{\phi}(0) + \frac{1}{2}\sum_{i=1}^{n_x} \frac{\partial^2\tilde{\phi}(x_k)}{\partial x_{i,k}^2}\bigg|_{x_k=0}, \tag{28}$$

where e_i is a Cartesian basis vector with all the elements equal to zero except the i-th term, which is unity.

Proof: The detailed proof is provided in Appendix A. ■

Recall that $\tilde{\phi}(x_k) = \phi(\mathcal{X}_k)$, where $\mathcal{X}_k = \hat{\mathcal{X}}_{k|k} + S_{k|k}x_k$ and $S_{k|k}$ is Cholesky factor of the covariance matrix $P_{k|k}$. Next, we transform the above partial derivatives of $\tilde{\phi}(x_k)$ into the desired partial derivatives of $\phi(\mathcal{X}_k)$, where as follows.

Proposition 2: The value of the coefficients A_k and B_k can be expressed as

$$A_k = [\nabla_{\mathcal{X}_k}\phi(\mathcal{X}_k)^T]^T|_{\mathcal{X}_k=\hat{\mathcal{X}}_{k|k}} S_{k|k} + \frac{1}{2}\sum_{i=1}^{n_x}\sum_{j=1}^{n_x} M(i,j)\,\phi(\mathcal{X}_k)|_{\mathcal{X}_k=\hat{\mathcal{X}}_{k|k}} e_i^T, \tag{29}$$

and

$$B_k = \phi(\hat{\mathcal{X}}_{k|k}) + \frac{1}{2}\sum_{i=1}^{n_x} e_i \mathrm{trace}[P_{k|k}[\nabla_{\mathcal{X}_k}\nabla^T_{\mathcal{X}_k}\phi(\mathcal{X}_k)]|_{\mathcal{X}_k=\hat{\mathcal{X}}_{k|k}}], \tag{30}$$

where we define the operator, $\nabla_{\mathcal{X}_k} = (\sum_{i=1}^{n_x} e_i \frac{\partial}{\partial\mathcal{X}_{i,k}})_{n_x\times 1}$, $M = S^T_{k|k}\nabla_{\mathcal{X}_k}\nabla^T_{\mathcal{X}_k}S_{k|k}diag(S^T_{k|k}\nabla_{\mathcal{X}_k})$.

Proof: The detailed proof is provided in Appendix B. ■

Remark 1: The coefficients of the measurement equation $C_k \in \mathbb{R}^{n_y\times n_x}$ and $D_k \in \mathbb{R}^{n_y}$ are also calculated similarly to A_k and B_k, and can be expressed as

$$C_k = [\nabla_{\mathcal{X}_{k+1}}\gamma(\mathcal{X}_{k+1})^T]^T\Big|_{\mathcal{X}_{k+1}=\hat{\mathcal{X}}_{k+1|k}} S_{k+1|k} + \frac{1}{2}\sum_{i=1}^{n_x}\sum_{j=1}^{n_x} M(i,j)\gamma(\mathcal{X}_{k+1})|_{\mathcal{X}_{k+1}=\hat{\mathcal{X}}_{k+1|k}} e_i^T, \tag{31}$$

$$D_k = \gamma(\hat{\mathcal{X}}_{k+1|k}) + \frac{1}{2}\sum_{i=1}^{n_y} e_i \mathrm{trace}[P_{k+1|k}[\nabla_{\mathcal{X}_{k+1}}\nabla^T_{\mathcal{X}_{k+1}}\gamma(\mathcal{X}_{k+1})]|_{\mathcal{X}_{k+1}=\hat{\mathcal{X}}_{k+1|k}}]. \tag{32}$$

Once A_k, B_k, C_k, D_k are evaluated at each iteration, an EKF like algorithm described in section III can be used to estimate the states. The steps to implement the Taylor series based EKF (TO-EKF) are outlined in Algorithm 1.

Algorithm 1: TO-EKF Algorithm

`Step 1: Initialization`
- Initialize the filter with $\hat{\mathcal{X}}_{0|0}$ and $P_{0|0}$.

`Step 2: Time update`
- Compute Cholesky decomposition: $P_{k|k} = S_{k|k}S^T_{k|k}$.
- Calculate A_k and B_k using Eqs. (29) – (30).
- Compute prior mean $\hat{\mathcal{X}}_{k+1|k} = B_k$.
- Calculate prior covariance $P_{k+1|k} = A_kA_k^T + Q_k$.

`Step 3: Measurement update`
- Compute $S_{k+1|k}$ such that $P_{k+1|k} = S_{k+1|k}S^T_{k+1|k}$.
- Evaluate C_k and D_k using Eqs. (31) – (32).
- Estimated value of the measurement $\hat{\mathcal{Y}}_{k+1|k} = D_k$.
- Calculate covariance of the measurement: $P^{\mathcal{YY}}_{k+1|k} = C_kC_k^T + R_{k+1}$.
- Calculate the cross-covariance between state and measurement: $P^{\mathcal{XY}}_{k+1|k} = S_{k+1|k}C_k^T$.
- Calculate the Kalman gain: $K_{k+1} = P^{\mathcal{XY}}_{k+1|k}(P^{\mathcal{YY}}_{k+1|k})^{-1}$.
- Determine the posterior estimate: $\hat{\mathcal{X}}_{k+1|k+1} = \hat{\mathcal{X}}_{k+1|k} + K_{k+1}(\mathcal{Y}_{k+1} - \hat{\mathcal{Y}}_{k+1|k})$.
- Estimate the posterior error covariance: $P_{k+1|k+1} = P_{k+1|k} - K_{k+1}P^{\mathcal{YY}}_{k+1|k}K^T_{k+1}$.

V. CALCULATION OF A_k, B_k, C_k AND D_k USING DIRECT NUMERICAL INTEGRATION

In this approach, the integral stated in Eq. (12) is evaluated with a few carefully chosen deterministic points and their corresponding weights [8], [9]. More clearly, for any real-valued function $f(x)$ which is integrable with respect to Gaussian measure, the integral (I) expressed below can be approximately evaluated as;

$$I = \int_{-\infty}^{\infty} f(x)w(x)dx \approx \sum_{i=1}^{m}\omega_i f(\xi_i), \tag{33}$$

where m is the total number of sample points, $w(x) = \mathcal{N}(x;0,I) = \frac{1}{(2\pi)^{n_x/2}}\exp(-\frac{1}{2}x^Tx)$ and (ξ_i,ω_i) are sample point-associated weight pairs. There are various methods available to generate these sample points and weights and evaluate such an integral in terms of an error controlled approximation. Spherical radial cubature rule [9], [10], Gauss-Hermite [8] and Monte Carlo integration [14] are some examples. The computation cost in Gauss Hermite and Monte Carlo based integration rule is high and increases exponentially with the increase in state dimension. To reduce the computation burden and yet to achieve comparable estimation accuracy here, we use third degree spherical radial cubature rule [9], [10] to evaluate the integral.

A. CUBATURE METHOD OF SAMPLE POINT GENERATION

In this method, the integral mentioned in Eq. (33) is decomposed into a surface and a line integral. Specifically, a third-degree spherical radial cubature rule is used to solve the surface integral, and the line integral is evaluated by the first-order Gauss-Laguerre quadrature rule. For a detailed derivation, the readers are referred to [9], [10]. Here we mention how to generate sample points and weights. The sample points generated through this method are called cubature points. The sample points are given as

$$\xi_i = \sqrt{2\lambda_{j'}}[u_j],$$

where $[u_j](j = 1, 2, \cdots, 2n_x)$ are located at the intersection of the unit sphere and its axes, and the quadrature points $\lambda_{j'}(j' = 1, 2, \cdots, n')$ are the roots of Chebyshev-Laguerre equation

$$\dot{L}^{\alpha}_{n'}(\lambda) = (-1)^{n'}\lambda^{-\alpha}\exp(\lambda)\frac{d^{n'}}{d\lambda^{n'}}\lambda^{\alpha+n'}\exp(-\lambda) = 0,$$

where $\alpha = (n_x/2 - 1)$. The weights can be given as

$$\omega_i = \frac{1}{2n_x\Gamma(n_x/2)}\frac{n'!\Gamma(\alpha+n'+1)}{\lambda_{j'}[\dot{L}^{\alpha}_{n'}(\lambda_{j'})]^2}, \tag{34}$$

where $i = 1, 2, \cdots, 2n_xn'$, and $j' = 1, 2, \cdots, n'$. Throughout the paper and during the simulation we assume $n' = 1$ and generate sample points.

Note that a higher degree cubature quadrature rule [11] can also be used, which might lead to somewhat higher accuracy at the expense of a higher computational burden.

B. EVALUATION OF A_k, B_k, C_k AND D_k

Whatever be the way we generate the sample points and weights, from Eq. (56), the coefficient A_k can be expressed as

$$A_k = \int_{-\infty}^{\infty}\phi(\hat{\mathcal{X}}_{k|k} + S_{k|k}x_k)x_k^T\mathcal{N}(x_k;0,I)dx_k. \tag{35}$$

With the help of the point-weight method as described in the previous subsection, the above expression can be written as

$$A_k = \sum_{i=1}^{m}\phi(\hat{\mathcal{X}}_{k|k} + S_{k|k}\xi_i)\xi_i^T\omega_i, \tag{36}$$

where ω_i is calculated using Eq. (34). Let $\chi_{i,k|k} = \hat{\mathcal{X}}_{k|k} + S_{k|k}\xi_i$, then the above equation becomes

$$A_k = \sum_{i=1}^{m}\phi(\chi_{i,k|k})\xi_i^T\omega_i. \tag{37}$$

In a similar fashion, B_k, C_k and D_k can be written as

$$B_k = \sum_{i=1}^{m}\phi(\chi_{i,k|k})\omega_i, \tag{38}$$

$$C_k = \sum_{i=1}^{m}\gamma(\chi_{i,k+1|k})\xi_i^T\omega_i, \tag{39}$$

$$D_k = \sum_{i=1}^{m}\gamma(\chi_{i,k+1|k})\omega_i, \tag{40}$$

where $\chi_{i,k+1|k} = \hat{\mathcal{X}}_{k+1|k} + S_{k+1|k}\xi_i$. The detailed algorithm to implement the CO-EKF is presented in the Algorithm 2.

Remark 2: The proposed CO-EKF is different from the CKF [9], [10]. In the proposed CO-EKF the coefficients of the first order polynomial are approximately evaluated through a set of cubature points and then the EKF is used for the system linearized using an orthogonal linear approximation. In contrast, in deterministic sample point filters, the mean and covariance are directly evaluated using deterministic sample points.

VI. AVOIDING THE NECESSITY OF THE SQUARE ROOT IMPLEMENTATION

In deterministic sample point filters, Cholesky decomposition is required to be performed at each step during which round-off error occurs due to limited arithmetic precision of the software [18], [19]. The numerical error accumulated over time leads the covariance matrix to lose positive definiteness, and Cholesky decomposition cannot be performed. The problem is well known and to circumvent it, square root implementation is required [9], [18]. Square root filtering increases the computational burden considerably. But the proposed method is free from this problem and the covariance matrices preserve the property of symmetry and positive definiteness during a software simulation.

From the expression of the prior error covariance in Eq. (20), it can be seen that the covariance matrix, $P_{k+1|k}$ preserves the property of symmetry and positive definiteness, and Cholesky decomposition can be calculated at each iteration of the measurement update step. The expression of posterior error covariance is

$$\begin{aligned} P_{k+1|k+1} &= P_{k+1|k} - K_{k+1}P^{\mathcal{Y}\mathcal{Y}}_{k+1|k}K^T_{k+1} \\ &= P_{k+1|k} - (K_{k+1}S^{\mathcal{Y}\mathcal{Y}}_{k+1|k})(K_{k+1}S^{\mathcal{Y}\mathcal{Y}}_{k+1|k})^T, \end{aligned} \tag{41}$$

which further can be expressed as [2, p. 206]

$$\begin{aligned} P_{k+1|k+1} = {} & (I - K_{k+1}C_{k+1}S^{-1}_{k+1|k})P_{k+1|k} \\ & \times (I - K_{k+1}C_{k+1}S^{-1}_{k+1|k})^T \\ & + K_{k+1}R_{k+1}K^T_{k+1}. \end{aligned} \tag{42}$$

Algorithm 2: CO-EKF Algorithm

`Step 1: Initialization`

- Generate the sample points ξ_i $(i = 1, 2, \cdots, 2n_x)$ and their corresponding weights ω_i using cubature rule.
- Initialize the filter with $\hat{\mathcal{X}}_{0|0}$ and $P_{0|0}$.

`Step 2: Time update`

- Compute the Cholesky decomposition of $P_{k|k} = S_{k|k}S_{k|k}^T$.
- Evaluate the cubature points $\chi_{i,k|k} = S_{k|k}\xi_i + \hat{\mathcal{X}}_{k|k}$.
- Propagated the cubature points $\chi^*_{i,k+1|k} = \phi(\chi_{i,k|k})$.
- Calculate A_k and B_k as follows:

$$A_k = \sum_{i=1}^{2n_x} \chi^*_{i,k+1|k}\xi_i^T\omega_i,\ B_k = \sum_{i=1}^{2n_x} \chi^*_{i,k+1|k}\omega_i.$$

- Estimate the prior mean $\hat{\mathcal{X}}_{k+1|k} = B_k$.
- Calculate the prior error covariance: $P_{k+1|k} = A_kA_k^T + Q_k$.

`Step 3: Measurement update`

- Factorize the prior error covariance: $P_{k+1|k} = S_{k+1|k}S_{k+1|k}^T$.
- Evaluate the cubature points: $\chi_{i,k+1|k} = S_{k+1|k}\xi_i + \hat{\mathcal{X}}_{k+1|k}$.
- Propagate the points through measurement: $y_{i,k+1|k} = \gamma(\chi_{i,k+1|k})$.
- Calculate C_k and D_k:

$$C_k = \sum_{i=1}^{2n_x} y_{i,k+1|k}\xi_i^T\omega_i,\ D_k = \sum_{i=1}^{2n_x} y_{i,k+1|k}\omega_i.$$

- Expected measurement: $\hat{\mathcal{Y}}_{k+1|k} = D_k$.
- Calculate the covariance of the measurement: $P^{\mathcal{Y}\mathcal{Y}}_{k+1|k} = C_kC_k^T + R_{k+1}$.
- The cross-covariance of the state and the measurement $P^{\mathcal{X}\mathcal{Y}}_{k+1|k} = S_{k+1|k}C_k^T$.
- Compute the Kalman gain $K_{k+1} = P^{\mathcal{X}\mathcal{Y}}_{k+1|k}(P^{\mathcal{Y}\mathcal{Y}}_{k+1|k})^{-1}$.
- Estimate the posterior state: $\hat{\mathcal{X}}_{k+1|k+1} = \hat{\mathcal{X}}_{k+1|k} + K_{k+1}(\mathcal{Y}_{k+1} - \hat{\mathcal{Y}}_{k+1|k})$.
- Estimate the posterior error covariance $P_{k+1|k+1} = P_{k+1|k} - K_{k+1}P^{\mathcal{Y}\mathcal{Y}}_{k+1|k}K_{k+1}^T$.

As mentioned in [2], the expression is less sensitive to the roundoff error and preserves symmetry as well as positive definiteness. This is a property inherited from the EKF, even though we are using a different form of linearization.

VII. COMPUTATIONAL COMPLEXITY

In this section, the computational complexity of the CKF, the square root CKF (SRCKF), and the proposed CO-EKF are calculated in terms of floating-point operations (flops) [30]. To compute the Cholesky decomposition of any matrix of order $n \times n$, $n^3/3 + 2n^2$ number of flops operation [30] is required. With this, we calculate the total flops count of a CKF with m sample point as

$$\begin{aligned} C(n_x,\ n_y,\ m) = {} & (6n_x^2 + (2 + n_y)2n_x + 3n_y + 2n_y^2)m \\ & + \frac{2}{3}n_x^3 + (7 + 2n_y)n_x^2 + (3 + 4n_x)n_y^2 \\ & + (1 + 2n_x)n_y + n_y^3, \end{aligned} \tag{43}$$

where n_x and n_y are the dimension of the state and the measurement, respectively and for the CKF, $m = 2n_x$. We count the flops required to implement the proposed CO-EKF as

$$\begin{aligned} & C(n_x,\ n_y,\ m) \\ & = (6n_x^2 + (3 + 2n_y)n_x + 3n_y)m + \frac{8}{3}n_x^3 \\ & \quad + 3n_x^2 + 2(2n_x - 1)n_xn_y + n_y + 6n_xn_y^2 + n_y^3. \end{aligned} \tag{44}$$

Now to calculate the flops count of the SRCKF, we need to calculate flops for QR decomposition. We assume that the Householder algorithm is used for it. For any matrix of order $m \times n$, the computational complexity of the Householder algorithm is $2mn^2 - 2/3n^3$ flops. With such QR decomposition, the total flops for the SRCKF are calculated as

$$\begin{aligned} C(n_x,\ n_y,\ m) = {} & (8n_x^2 + (3 + 2n_y)2n_x + 4n_y + 2n_y^2)m \\ & + 2n_x^3 + 2n_x^2n_y + 4n_xn_y^2 + 2n_y^3 + n_y. \end{aligned} \tag{45}$$

We compare the flops count of the filtering algorithm by varying state dimension, n_x for several fixed n_y/n_x, such as 1/3, 1/4 and 1/20, and plotted in Figs. 1a - 1c. From the figures, it can be seen that the flops count of the SRCKF is (always) higher than the proposed CO-EKF. At a lower value of n_x, the flops count of the CKF, the SRCKF, and the CO-EKF seem to coincide. To illustrate this, now we take a smaller value of n_x such as 5, and vary the n_y/n_x from 0.2 to 1, and the flops count vs. n_y/n_x is plotted in Fig. 1d. From the figure, we can see that the flops count of the SRCKF is much higher than the CO-EKF, particularly when the dimension of the measurement vector approaches the state vector's dimension.

It can be noted that while calculating the cumulative computational complexity, we have not accounted the function evaluation related to the state or measurement equation. Due to this reason, the actual time of the computation differs from the flops count. Moreover, the specific function evaluation of a given state-space model can be easily accounted in flops.

VIII. FILTERING PERFORMANCE METRICS

In this section, we briefly discuss the performance metrics which we use to compare the performances of various filtering methods.

A. ROOT MEAN SQUARE ERROR

The absolute filtering performance or estimation accuracy of the filters can be evaluated by the root mean square error (RMSE), which can be calculated at k-th time step

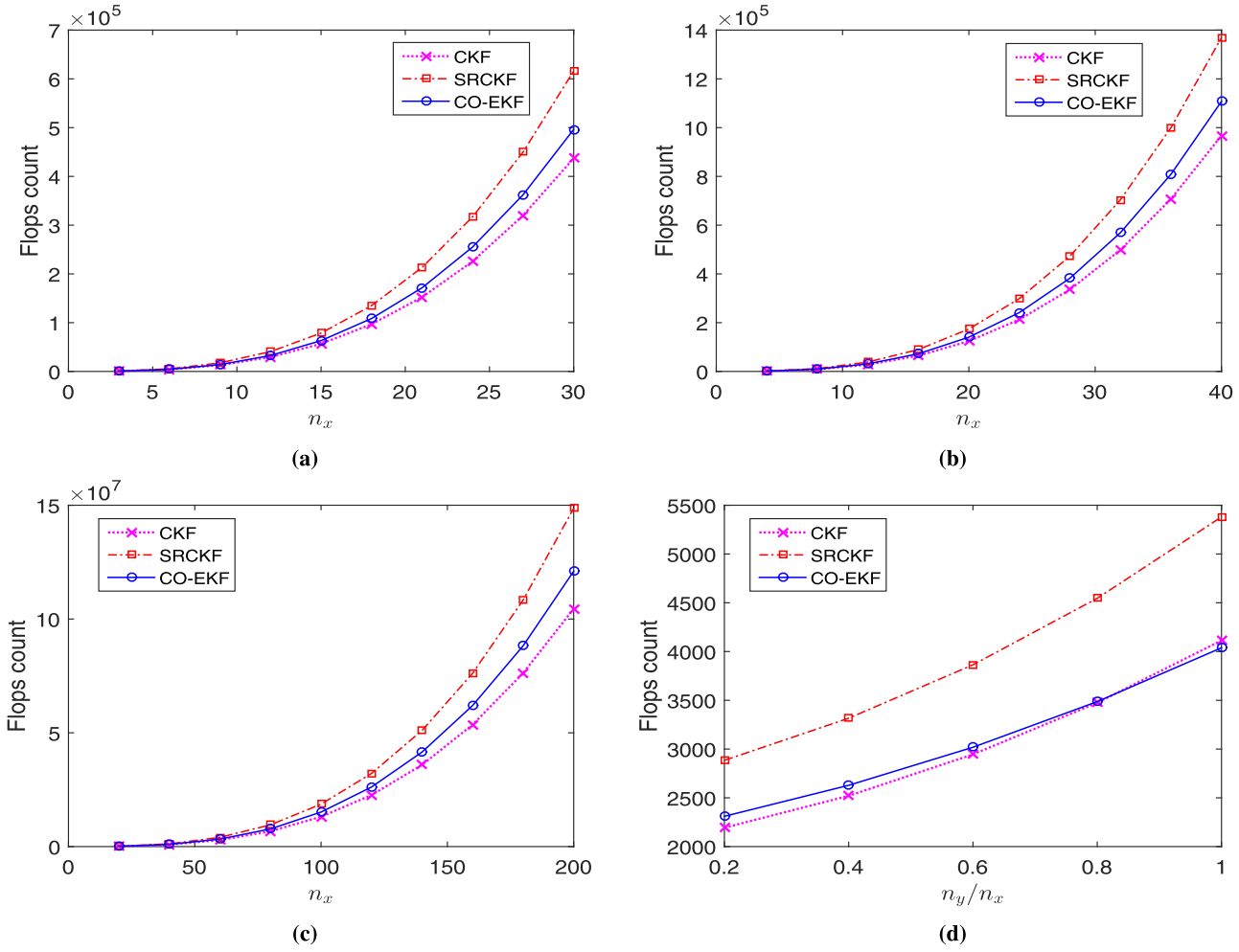

FIGURE 1. Flops count vs. state dimension (n_x) plot of the CKF, SRCKF and CO-EKF when (a) $n_y/n_x = 1/3$, where n_y varies from 1 to 10, and n_x varies from 3 to 30; (b) $n_y/n_x = 1/4$, where n_y varies from 1 to 10, and n_x varies from 4 to 40; (c) $n_y/n_x = 1/20$, where n_y varies from 1 to 10, and n_x varies from 20 to 200; (d) the flops count vs. n_y/n_x plot for $n_x = 5$, and n_y varies from 1 to 5. Although the CKF takes the lowest flops count to execute, it is not free from Cholesky decomposition error. Both the proposed CO-EKF and SRCKF are free from Cholesky decomposition error and among them the CO-EKF has the lowest flops count.

as [2, p. 243]

$$\text{RMSE}_k = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\mathcal{X}_{i,k} - \hat{\mathcal{X}}_{i,k})^2},$$

where M is the total number of Monte Carlo (MC) runs, $\mathcal{X}_{i,k}$ and $\hat{\mathcal{X}}_{i,k}$, are the truth and estimated state at the k-th time-step of i-th Monte Carlo runs.

B. NORMALIZED (STATE) ESTIMATION ERROR SQUARED

The consistency of a filter during estimation is crucial to practitioners, and it can be evaluated by the normalized estimation error squared (NEES). The NEES at k-th time-step is defined as [2, p. 234] [31]

$$\text{NEES}_k = (\mathcal{X}_k - \hat{\mathcal{X}}_{k|k})^T P_{k|k}^{-1}(\mathcal{X}_k - \hat{\mathcal{X}}_{k|k}),$$

where $\mathcal{X}_k$ is the truth, $\hat{\mathcal{X}}_{k|k}$ is the posterior state estimate, and $P_{k|k}$ is the posterior error covariance. The average NEES (ANEES) for M Monte Carlo runs can be defined as [2, p. 234]

$$\text{ANEES}_k = \frac{1}{M}\sum_{i=1}^{M}\text{NEES}_k^i.$$

The filter is considered to consistent [2, p. 235] if $\text{ANEES}_k \in [l_b, u_b]$, where l_b and u_b are lower and upper bound, respectively. For 95 % probability region, l_b and u_b are computed as follows [31]:

$$l_b = n_x\Big[\Big(1 - \frac{2}{9n_xM}\Big) - 1.96\sqrt{\frac{2}{9n_xM}}\Big]^3, \tag{46}$$

$$u_b = n_x\Big[\Big(1 - \frac{2}{9n_xM}\Big) + 1.96\sqrt{\frac{2}{9n_xM}}\Big]^3. \tag{47}$$

If the $\text{ANEES}_k < l_b$, then filter is considered 'pessimist' ('under confident'), since the posterior error covariance is very high compare to its true value [2, p. 245]. On the

contrary, if $\text{ANEES}_k > u_b$ then filter is said to be 'optimistic' ('over confident'), since the covariance ($P_{k|k}$) is too small [2, p. 245].

C. BIAS NORM

The bias norm [32] of a filter obtained from M Monte Carlo runs, can be evaluated as

$$\text{Bias norm}_k = \left\| \frac{1}{M}\sum_{i=1}^{M} \hat{\mathcal{X}}_k^i - \frac{1}{M}\sum_{i=1}^{M} \mathcal{X}_k^i \right\|_2,$$

where the norm of any vector $a = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}^T$, is given by $\|a\|_2 = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$.

D. FAIL COUNT OR TRACK LOSS

A fail count or track loss situation occurs in filtering when the absolute estimation error fails to settle below the specified limit, denoted by e_b. In such a case, the state estimate deviates from its path without following the real truth state. We define the filter fails to track in that MC runs when $\mathcal{X}_{lt}^i - \hat{\mathcal{X}}_{lt}^i > e_b$, where $\mathcal{X}_{lt}^i$ and $\hat{\mathcal{X}}_{lt}^i$ are the truth and estimated state at last time-step of i-th Monte Carlo runs.

IX. SIMULATION RESULTS

Problem 1: Here we consider a single-dimensional system [10] with process equation:

$$\mathcal{X}_{k+1} = \mathcal{X}_k + t\phi(\mathcal{X}_k) + \eta_k,$$

and the measurement equation:

$$\mathcal{Y}_{k+1} = t\gamma(\mathcal{X}_{k+1}) + \nu_{k+1},$$

where $\phi(\mathcal{X}_k) = 5\mathcal{X}_k(1 - \mathcal{X}_k^2)$, $\gamma(\mathcal{X}_k) = \mathcal{X}_k(1 - 0.5\mathcal{X}_k)$, $\eta_k \sim \mathcal{N}(0, Q_k)$, and $\nu_k \sim \mathcal{N}(0, R_k)$. During simulation, we use $b = 0.5$, $d = 0.1$, $t = 0.01$ sec, $Q_k = b^2\, t$ and $R_k = d^2\, t$. We assume the initial state of the system is $\mathcal{X}_0 = -0.2$, the initial posterior state estimate is $\hat{\mathcal{X}}_{0|0} = 0.8$ and the initial posterior error covariance is $P_{0|0} = 2$. The estimation is done for 4 sec. The system has two stable equilibrium points at 1 and -1, and one unstable equilibrium point at 0. Even for a moderate estimation error, the estimated state settles at wrong equilibrium points, and a fail count situation occurs [10].

The state is estimated by the EKF, the CKF, the CO-EKF, and the TO-EKF. For a single representative run, we plot the truth and estimated state obtained from different filters in Fig. 2a. From the figure, we see that the CKF, the CO-EKF, and the TO-EKF follow the truth, whereas the EKF loses track. It happens due to a large estimation error of the EKF forces to settle the estimate to a wrong equilibrium point. The filtering performance has been compared in terms of root mean square error (RMSE) obtained from 1000 MC runs and is shown in Fig. 2b. From the figure, it can be seen that the TO-EKF provides a better result than the EKF. The CKF and the proposed CO-EKF provide almost identical results. We also compare the filtering performance in terms of average NEES, and average NEES plot of the different filters excluding the failed runs (with $e_b = 1$) is shown in Fig. 2c. Further, we have plotted the bias norm of different filters vs. time in Fig. 2d. The Fig. 2d shows that the EKF has the largest bias norm, whereas the TO-EKF retains the lowest. The CKF and the CO-EKF show almost similar performance.

Lastly, we compare the filtering performance in terms of fail counts. The percentage fail counts obtained from 1000 MC runs are presented in Table 1. From the table, we see that the EKF has the highest fail counts (because frequently it settles to a wrong equilibrium point), whereas the TO-EKF has the lowest. The fail counts of the CKF and CO-EKF are almost similar. As it is a single-dimensional problem, any difference in execution time is quite modest and is not reported.

TABLE 1. Percentage fail counts of different filters obtained from 1000 MC runs.

Filter	Fail count (%)
EKF	23.6
CKF	6
CO-EKF	6.2
TO-EKF	3.5

Problem 2: In this example, we consider a three-dimensional Lorentz system [8] with the following state space model

$$\mathcal{X}_{k+1} = \phi(\mathcal{X}_k) + b\eta_k,$$
$$\mathcal{Y}_k = \gamma(\mathcal{X}_k) + d\nu_k,$$

where $\phi(\mathcal{X}_k) = \mathcal{X}_k + \delta t f(\mathcal{X}_k)$ and $\gamma(\mathcal{X}_k) = \delta t h(\mathcal{X}_k)$. Here, $\mathcal{X}_k \in \mathbb{R}^3$ is the state of the system, and $\mathcal{Y}_k \in \mathbb{R}$ is the measurement equation. The function $f(\mathcal{X})$ and $h(\mathcal{X})$ are given as

$$f(\mathcal{X}) = [r_1(-\mathcal{X}_1 + \mathcal{X}_2),\ r_2\mathcal{X}_1 - \mathcal{X}_2 - \mathcal{X}_1\mathcal{X}_3,\ -r_3\mathcal{X}_3 + \mathcal{X}_1\mathcal{X}_2]^T,$$

and

$$h(\mathcal{X}) = \sqrt{(\mathcal{X}_1 - 0.5)^2 + \mathcal{X}_2^2 + \mathcal{X}_3^2}.$$

η_k and ν_k are uncorrelated white Gaussian noises with zero mean and covariance $\delta t = 0.01$. The parameters r_1, r_2 and r_3 are known as Prandtl number, Rayleigh number and geometric factor, respectively. In this problem, we choose $r_1 = 10, r_2 = 28$ and $r_3 = 8/3$ [8], [10]. The value of $r_2 = 28$ is more than 24.74 which means that the system is unstable and the trajectory is quite sensitive to the initial condition. The values of system parameters have been taken as $b = \begin{bmatrix} 0 & 0 & 5 \end{bmatrix}^T$ and $d = 0.2$. The system is initialized with $\mathcal{X}_0 = \begin{bmatrix} -0.2 & -0.3 & -0.5 \end{bmatrix}^T$. The filter is initialized with $\hat{\mathcal{X}}_{0|0} = \begin{bmatrix} 1.35 & -3 & 6 \end{bmatrix}^T$ and with the initial covariance $P_{0|0} = 0.35\, I$. Estimation is performed for the time span of 0 to 4 sec.

We implemented the EKF, the CKF, the CO-EKF and the TO-EKF, and the truth vs. estimated state (TO-EKF)

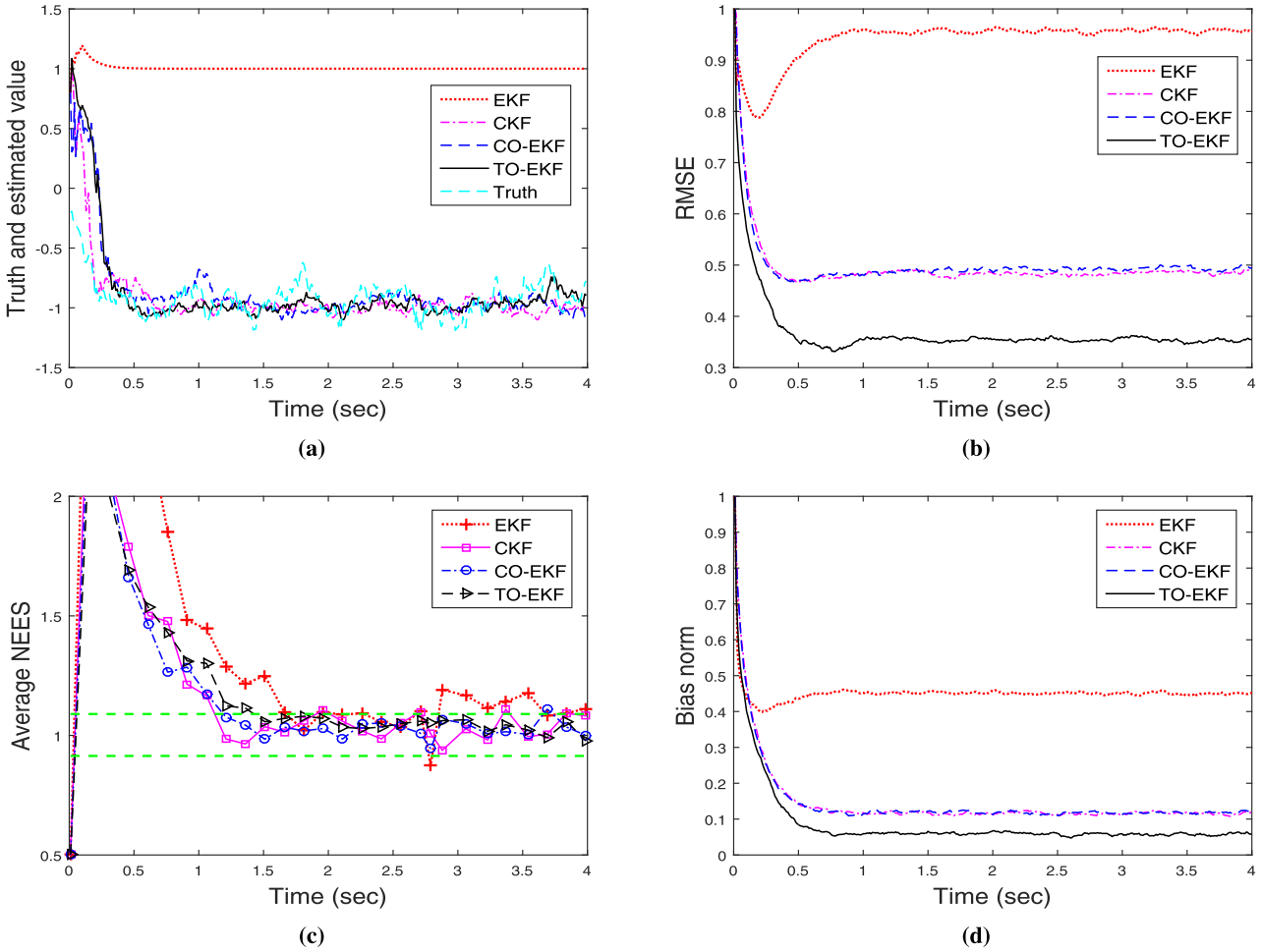

FIGURE 2. (problem 1): (a) The truth and the estimated values (obtained from the EKF, CKF, CO-EKF, and TO-EKF) for a single representative run; (b) the RMSE vs. time plot for the EKF, CKF, CO-EKF, and TO-EKF obtained from 1000 MC runs; (c) the average NEES of different filters vs. time plot obtained from 1000 MC runs; (d) the bias norm of the EKF, CKF, CO-EKF and TO-EKF vs. time plot obtained from 1000 MC run.

plot is shown in Fig. 3a. During the implementation of the CKF, we encountered the Cholesky decomposition error, so we use the square root CKF (SRCKF) [18]. The filtering performance is compared in terms of RMSE obtained from 100 MC runs. The RMSE of three states obtained from 100 Monte Carlo runs are plotted in Figs. 3b–3d. From the figures, it can be observed that for state-1 and state-2, the TO-EKF provides a more accurate estimation than the EKF. The SRCKF and the CO-EKF results are comparable.

We compare the consistency and bias of the filters in terms of average NEES and bias norm, respectively. The average NEES and bias norm of the different filters are calculated, excluding failed trajectories. We define an estimator fails to track when $\sum_{k=100}^{400}(\mathcal{X}_{1,k} - \hat{\mathcal{X}}_{1,k})^2 > 10^4$. Fig. 4a shows the average NEES vs. time plot. From the figure, it can be observed that the average NEES of the SRCKF, the CO-EKF, and the TO-EKF are within the lower and upper bounds which are $l_b = 2.5391$ and $u_b = 3.4988$, respectively. However, the EKF does not lie within the bounds. The bias norm of different filters is shown in Fig. 4b. From the figure, we see that the EKF has the highest bias norm, whereas the SRCKF, the CO-EKF, and the TO-EKF provide similar results which are lower than the values obtained from the EKF. The relative execution time of different filters with respect to the EKF are reported in Table 2. From the table, it can be seen that the SRCKF demands almost double computational time than the EKF, however, the CO-EKF and the TO-EKF take nearly 20% less execution time than the SRCKF.

TABLE 2. The relative execution time taken by different filters.

Filter	Execution time
EKF	1.00
SRCKF	2.08
CO-EKF	1.63
TO-EKF	1.61
CKF	1.59

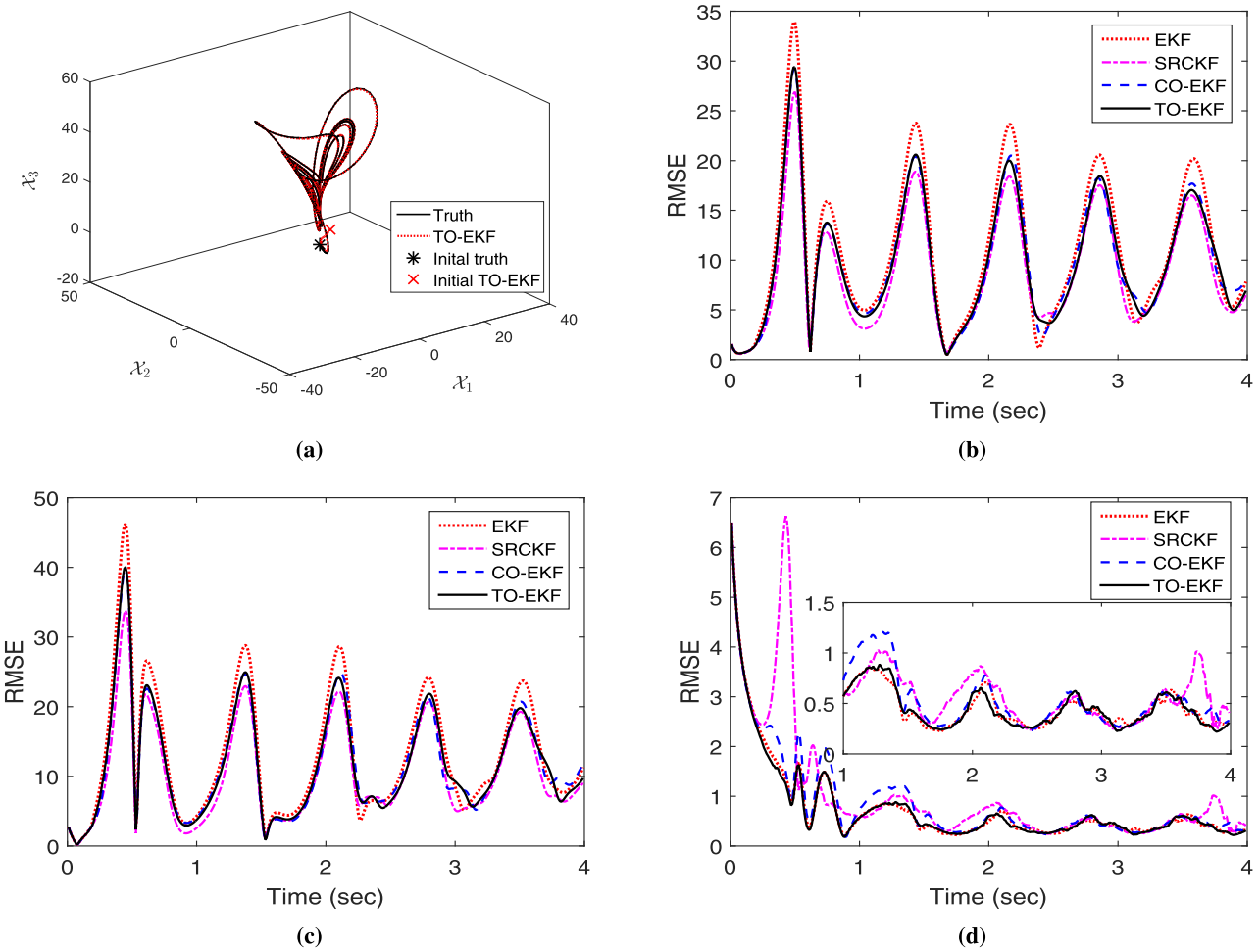

FIGURE 3. (problem 2): (a)The truth and estimated state of the proposed TO-EKF for a single run; (b) the RMSE of first state; (c) the RMSE of second state; (d) the RMSE of third state of the EKF, SRCKF, CO-EKF, and TO-EKF for 100 MC runs. The SRCKF and CO-EKF have used $2n_x$ *i.e.* 6 sample points. Further, in the TO-EKF, a third-order Taylor series approximation is used.

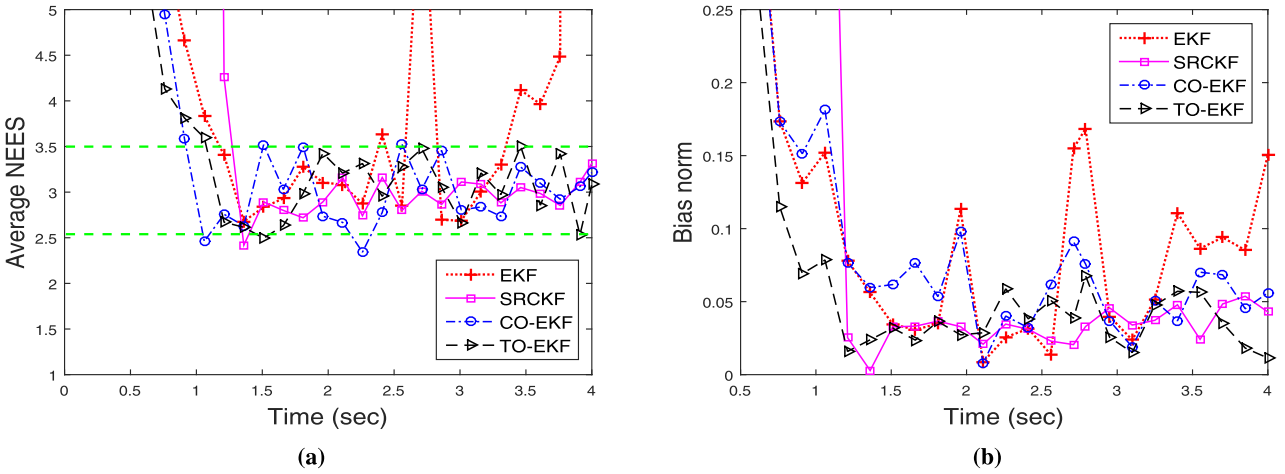

FIGURE 4. (problem 2): (a) Average NEES vs. time plot; (b) bias norm vs. time plot for the EKF, SRCKF, CO-EKF, and TO-EKF for 100 MC runs.

Problem 3: In this example, a bearing only tracking (BOT) problem [33] has been considered where a moving target is being tracked. The target and platform kinematics are shown in Fig. 5.

Process model: The target dynamics on the X-axis in discrete-time is

$$\mathcal{X}_{k+1} = FX_k + \Gamma_k \eta_k, \tag{48}$$

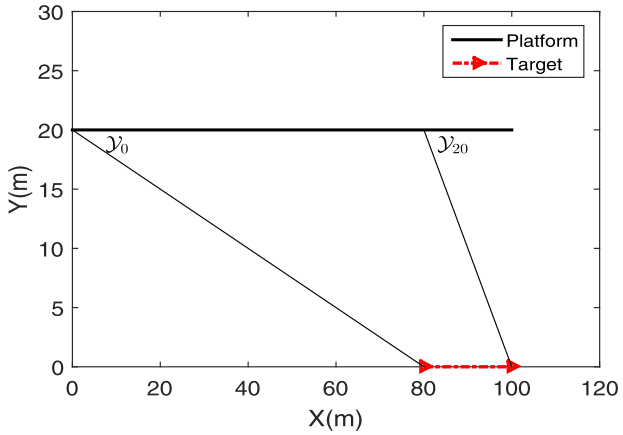

FIGURE 5. (problem 3): A target (ship) is moving on a straight line and an observer (aero platform) passing on the top of it and measuring bearing angles.

where $\mathcal{X}_k = \begin{bmatrix}\mathcal{X}_{1,k}\\ \mathcal{X}_{2,k}\end{bmatrix}$, $F = \begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix}$, $\Gamma_k = \begin{bmatrix}T^2/2\\ T\end{bmatrix}$ with $\mathcal{X}_{1,k}$ and $\mathcal{X}_{2,k}$ are the position and velocity along the X-axis, respectively. The process noise η_k is white Gaussian with mean zero and covariance $q_k = 0.01\text{m}^2/\text{sec}^4$.

Observation model: The platform dynamics in discrete-time domain may be represented as

$$x_{p,k} = \bar{x}_{p,k} + \Delta x_{p,k} \quad k = 1, 2, \cdots, n_{\text{step}}, \tag{49}$$
$$y_{p,k} = \bar{y}_{p,k} + \Delta y_{p,k} \quad k = 1, 2, \cdots, n_{\text{step}}, \tag{50}$$

where $\bar{x}_{p,k}$ and $\bar{y}_{p,k}$ are the average platform position coordinates, $n_{\text{step}} = 20$, $\Delta x_{p,k}$ and $\Delta y_{p,k}$ are assumed to be white Gaussian and mutually independent noises with covariances $r_x = 1\text{m}^2$ and $r_y = 1\text{m}^2$, respectively. The average platform position co-ordinates are $\bar{x}_{p,k} = 4kT$ and $\bar{y}_{p,k} = 20$, where $T = 0.2$ sec is sampling time. The bearing measurement is represented as

$$\mathcal{Y}_k = \gamma[x_{p,k}, y_{p,k}, \mathcal{X}_{1,k}] + v_{s,k} = \tan^{-1}\left(\frac{y_{p,k}}{\mathcal{X}_{1,k} - x_{p,k}}\right) + v_{s,k}.$$

The measurement noise, $v_{s,k}$ is white Gaussian with zero mean and covariance $r_s = (3^\circ)^2$. The random platform perturbation induces extra error in measurement. Combining all these uncertainties, the measurement equation will be approximated as

$$\mathcal{Y}_k \approx \gamma[\bar{x}_{p,k}, \bar{y}_{p,k}, \mathcal{X}_{1,k}] + v_k, \tag{51}$$

where v_k is the equivalent measurement noise with mean zero and covariance R_k given by [33],

$$E[v_k^2] = R_k = \frac{\bar{y}_{p,k}^2 r_x + [\mathcal{X}_{1,k} - \bar{x}_{p,k}]^2 r_y}{\{[\mathcal{X}_{1,k} - \bar{x}_{p,k}]^2 + \bar{y}_{p,k}^2\}^2} + r_s.$$

The EKF, the SRCKF, the CO-EKF and the TO-EKF are implemented on the problem described above. The initial truth of the state is $\mathcal{X}_0 = [80\ \ 1]^T$ and we initialize the estimators with the initial state estimate ($\hat{\mathcal{X}}_{0|0}$) and the error covariance ($P_{0|0}$) as discussed in [33]. For a single representative run, the truth vs. estimated state (TO-EKF) plot for position and velocity are shown in Fig. 6a and Fig. 6b, respectively. Filtering performance is compared in terms of RMSEs, which are calculated over 100 MC runs.

The RMSE of position and velocity are plotted in Fig. 6c and Fig. 6d, respectively. Please note that the RMSEs are calculated excluding track loss cases which is defined when terminal position error goes beyond 15 m. From the figures, it can be seen that the RMSEs for both position and velocity are almost similar for all filters and somewhat better than the EKF. We also compare the filtering performance in terms of the average NEES and the bias norm. For the 95% probability concentration region, we calculate $l_b = 1.8285$ and $u_b = 2.1791$ using Eqs. (46)–(47). The average NEES of the EKF, the SRCKF, the CO-EKF, and the TO-EKF are plotted in Fig. 7a. From the figure, we see that the average NEES of the SRCKF, the CO-EKF, and the TO-EKF lie in the concentration region whereas the EKF has a high average NEES value and does not lie in the defined region. Fig. 7b compares the bias for all the filters. From the figure, we see that the SRCKF, the CO-EKF, and the TO-EKF have similar bias norms, and only the EKF performs poorly.

Lastly, the filters are compared in terms of track loss and execution time and are shown in Table 3. We calculate the percentage of track loss from one hundred thousand MC runs. From Table 3, it can be seen that the EKF has the highest track loss, followed by the TO-EKF. The track loss of the CO-EKF and the SRCKF are both very low and are nearly the same. However, the execution time of the SRCKF is higher than the CO-EKF.

TABLE 3. Percentage track loss and relative execution time taken by different filters.

Filter	Track loss (%)	Execution time
EKF	0.163	1.00
SRCKF	0.007	1.40
CO-EKF	0.009	1.20
TO-EKF	0.021	1.14
CKF	0.007	1.17

Problem 4: In this problem, we consider a real-life passive underwater bearing-only tracking (BOT) problem [32], [34]. It is assumed that the target moves with a constant velocity and the observer follows a maneuvered path. The target movement and observer maneuvering are presented in Fig. 8. Our main objective is to estimate the trajectory of the target *i.e.* position and velocity of the target from the noisy sensor measurement. We denote the target state variable as $\mathcal{X}^t = \left[\mathcal{X}_1^t\ \mathcal{X}_2^t\ \dot{\mathcal{X}}_1^t\ \dot{\mathcal{X}}_2^t\right]^T$, and the observer states by $\mathcal{X}^o = \left[\mathcal{X}_1^o\ \mathcal{X}_2^o\ \dot{\mathcal{X}}_1^o\ \dot{\mathcal{X}}_2^o\right]^T$. We define a relative state vector as $\mathcal{X} = \mathcal{X}^t - \mathcal{X}^o = \left[\mathcal{X}_1\ \mathcal{X}_2\ \dot{\mathcal{X}}_1\ \dot{\mathcal{X}}_2\right]^T$, and the dynamic model (process equation) in discrete-time domain can be represented as

$$\mathcal{X}_{k+1} = F\mathcal{X}_k - u_{k,k+1} + \eta_k, \tag{52}$$

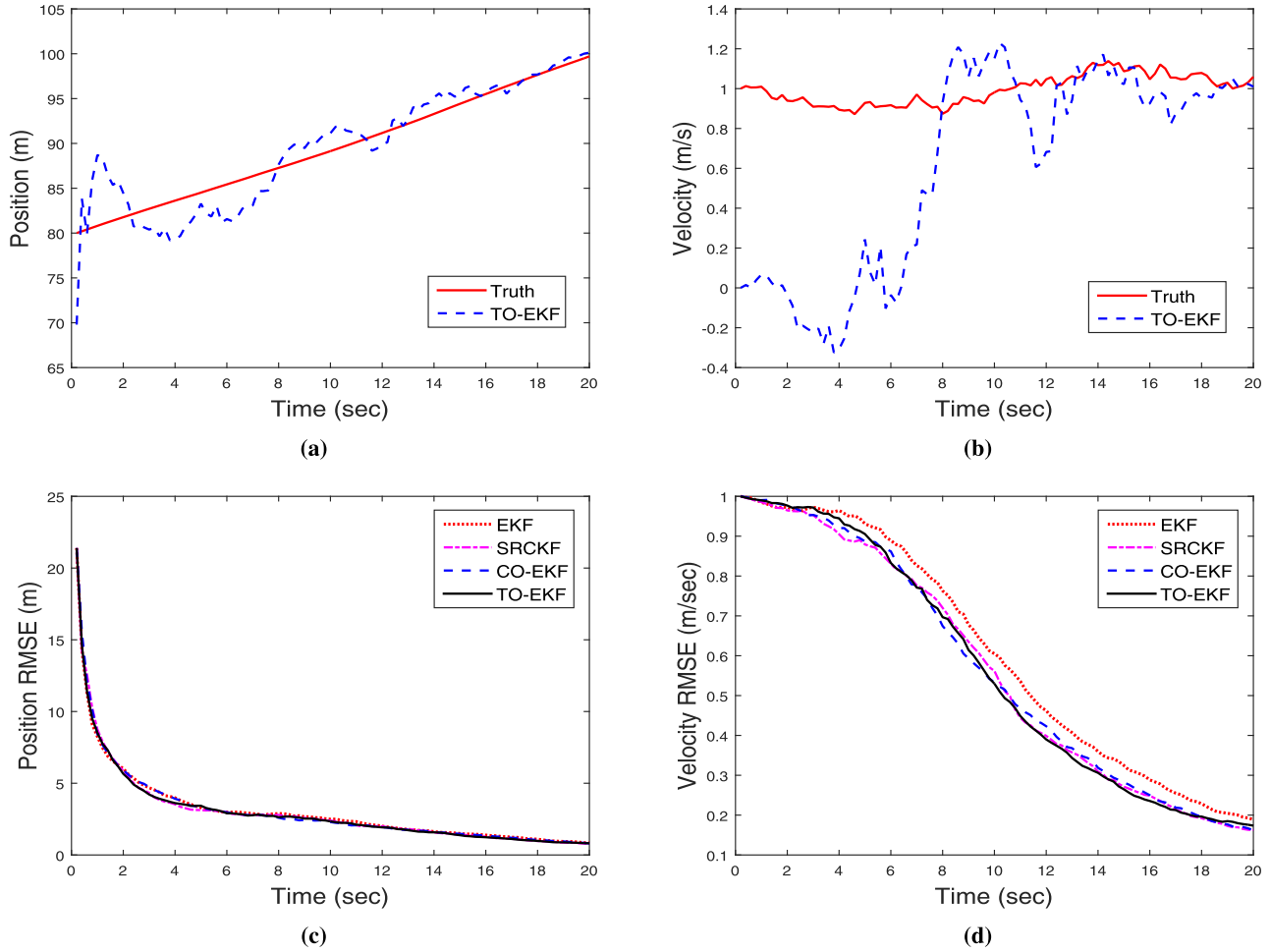

FIGURE 6. (problem 3): The truth and the estimated value of (a) position; (b) velocity obtained from the proposed TO-EKF for a single representative run; (c) position RMSE vs. time plot; (d) velocity RMSE vs. time plot of the EKF, SRCKF, CO-EKF, and TO-EKF for 500 MC runs. The number of required sample points to implement SRCKF and CO-EKF are $2n_x$ *i.e.* 4, and a third-order Taylor series approximation is used during the implementation of TO-EKF.

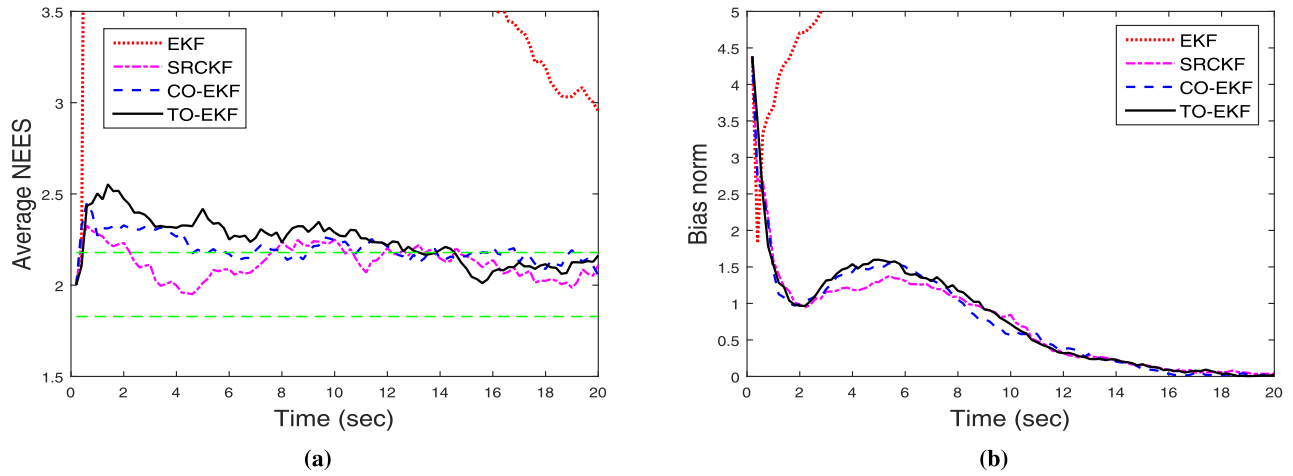

FIGURE 7. (problem 3): Comparison of the EKF, SRCKF, CO-EKF, and TO-EKF in terms of (a) average NEES; (b) bias norm for 500 MC runs.

where the state transition matrix is

$$F = \begin{bmatrix} I_{2\times 2} & TI_{2\times 2} \\ 0_{2\times 2} & I_{2\times 2} \end{bmatrix}.$$

The effects of observer acceleration is accounted in terms of deterministic input $u_{k,k+1}$, which is defined as $u_{k,k+1} = [\mathcal{X}^o_{1,k+1} - \mathcal{X}^o_{1,k} - T\dot{\mathcal{X}}^o_{1,k}, \quad \mathcal{X}^o_{2,k+1} - \mathcal{X}^o_{2,k} - T\dot{\mathcal{X}}^o_{2,k},$

$\dot{\mathcal{X}}^o_{1,k+1} - \dot{\mathcal{X}}^o_{1,k}, \quad \dot{\mathcal{X}}^o_{2,k+1} - \dot{\mathcal{X}}^o_{2,k}]^T$. The process noise is Gaussian with mean zero and covariance

$$Q_k = q \begin{bmatrix} \frac{T^3}{3} I_{2\times 2} & \frac{T^2}{2} I_{2\times 2} \\ \frac{T^2}{2} I_{2\times 2} & T I_{2\times 2} \end{bmatrix}.$$

The available noisy sensor measurement is given by

$$\mathcal{Y}_k = \gamma(\mathcal{X}_k) + \nu_k,$$

where $\nu_k \sim \mathcal{N}(0, \sigma_\theta^2)$. The true bearing is given by $\gamma(\mathcal{X}_k) = \text{atan2}(\mathcal{X}_{1,k}, \mathcal{X}_{2,k})$, and measured it from the observer to target in clockwise direction with respect to true north. The parameters used in the simulation are tabulated in Table 4, and the filtering run time is 30 min. The filters are initialized as described in [32].

TABLE 4. The parameters used in tracking scenario.

Parameters	Values
Initial range (r)	5 km
Target speed (s)	4 knots
Target course	-140°
Observer speed	5 knots
Observer initial course	140°
Observer final course	20°
Observer maneuver	From 13th to 17th min
σ_θ	1.5°
q	1.944×10^{-6} km^2/min^3

In this problem, we implemented the EKF, the SRCKF, the CO-EKF, and the TO-EKF to track the target trajectories. In Fig. 8, we show the target tracking using the proposed TO-EKF. From the figure, we see that the TO-EKF starts converging after maneuvering (please note the maneuvering point in both own ship and estimated trajectory) takes place. We compare the performance of the filters in terms of position

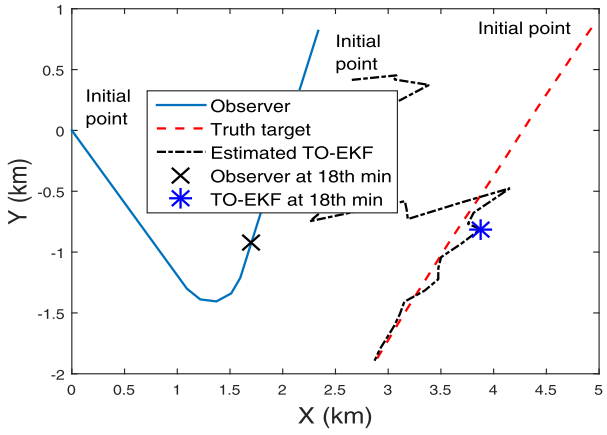

FIGURE 8. (problem 4): The figure shows a typical engagement scenario where a target moves in a straight line, and the observer ship is maneuvering to track the target. The estimation result shows TO-EKF successfully tracks the target although the initial error is high.

and velocity RMSE obtained from $M = 500$ MC runs. The position RMSE at k-th time-step obtained from M MC runs can be expressed as

$$\text{pos}_k = \sqrt{\frac{1}{M}\sum_{i=1}^{M} (\mathcal{X}^i_{1,k} - \hat{\mathcal{X}}^i_{1,k})^2 + (\mathcal{X}^i_{2,k} - \hat{\mathcal{X}}^i_{2,k})^2}.$$

The position and velocity RMSE vs. time (excluding track loss) are plotted in Fig. 9a and Fig. 9b, respectively. We define the track loss when the terminal position error goes beyond 1 Km (*i.e* e_b =1Km). From the Figs. 9a–9b, we see that the proposed filters are better than the traditional EKF, and the TO-EKF is slightly better among them.

Now, we check the filtering consistency using average NEES. For 95% probability concentration region, we calculate the values of lower bound $l_b = 3.7559$ and upper bound $u_b = 4.2517$ using Eq. (46) and Eq. (47), respectively. We have plotted the average NEES of different estimators in Fig. 10a considering the runs which achieve terminal position error less than 150 m. From the figure, we see that only the TO-EKF could reach within the bounds and all other filters lie outside the bound. We also compared the filters in terms of bias norm plotted in Fig. 10b. It can be seen that the CO-EKF has the lowest bias norm, whereas the EKF has the highest one.

Lastly, we compare the filters in terms of track loss and execution time which are shown in Table 5. The track loss (for $e_b = 1$km) has been calculated from 100 thousands MC runs. Table 5 shows that the track loss of the EKF is the highest, whereas the SRCKF has the lowest. The track loss of the CO-EKF is almost near to the SRCKF. However, the execution time of the SRCKF is almost 1.5 times of the CO-EKF.

TABLE 5. Percentage track loss when error bound is 1 km for 100 thousand MC runs and relative execution time taken by different filters.

Filter	Track loss (%)	Execution time
EKF	5.032	1.00
SRCKF	2.046	2.17
CO-EKF	2.290	1.43
TO-EKF	2.732	1.68
CKF	2.046	1.82

Based on the four simulation examples, we can draw the following broad qualitative conclusions:

(i) Both the TO-EKF and the CO-EKF consistently outperform the traditional EKF.

(ii) Both the new filters have accuracy comparable with that of the square root form of an integration-based filter, and the CO-EKF also has track loss performance which is comparable with the SRCKF. While the tracking example shows the track loss of the TO-EKF to be higher than that of the SRCKF, it is still less than the EKF.

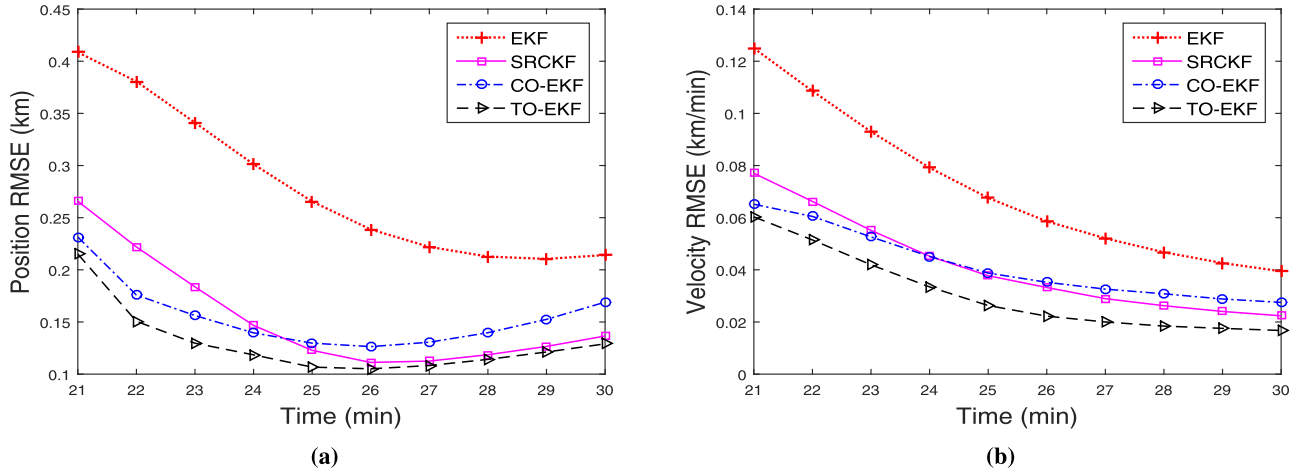

FIGURE 9. (problem 4): (a) The position RMSE vs. time plot; (b) the velocity RMSE vs. time plot of the EKF, SRCKF, CO-EKF, and TO-EKF for 500 MC runs. The number of required sample points to implement the SRCKF and CO-EKF is $2n_x$ *i.e.* 8, and third-order Taylor series approximation is used in the implementation of the TO-EKF.

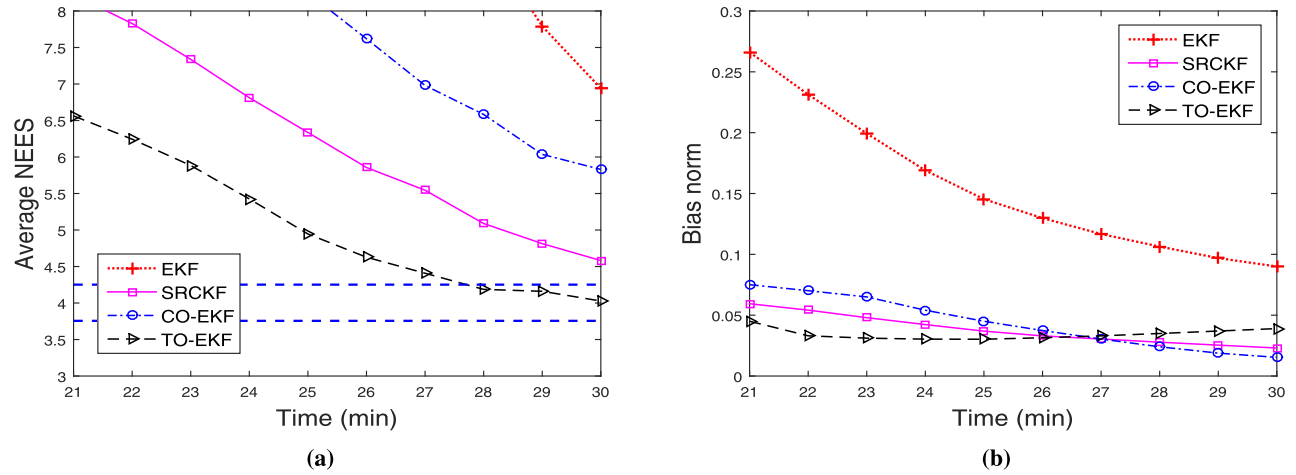

FIGURE 10. (problem 4): (a) The average NEES vs. time plot calculated from 500 MC runs; (b) The bias norm vs. time plot of different filters obtained from 500 MC runs.

(iii) Both the TO-EKF and the CO-EKF have execution time that is higher than the EKF but below that of the SRCKF.

(iv) Comparison with other numerical integration-based filters such as the Gauss Hermite filter and the unscented Kalman filter leads to the same or similar qualitative conclusions: the proposed filters yield comparable accuracy to integration-based filters at a lower computational cost. Results of numerical comparison with other filters are omitted for brevity and clarity of exposition.

X. DISCUSSION AND CONCLUSION

This paper proposes a new approach that is based on linearization of the process and measurement equation with first-order orthogonal Hermite polynomial, to solve a nonlinear state estimation problem. The developed estimation method is expected to perform more accurately than the Taylor series-based linearization. The coefficients of the first order polynomial are approximately calculated by evaluating the integrals using two different alternatives: (i) third-order Taylor series expansion of the function (ii) cubature integration method.

Since we are focused on recursively computing moments with respect to the Gaussian measure, using an optimal linear approximation (in integral squared error sense) under Gaussian measure sounds like an intuitively more attractive choice than using Taylor series based linearization. Four different simulation examples show that the new filters consistently outperform the EKF and give accuracy which is comparable with computationally more expensive square root filtering, at a lower computational cost.

APPENDIX A

Proof of Proposition 1: To calculate the coefficients A_k and B_k, at first, we approximate the function $\tilde{\phi}(x_k)$ by a

third-order Taylor series:

$$\tilde{\phi}(x_k) \approx \tilde{\phi}(0) + \sum_{i=1}^{n_x} x_{i,k} \left.\frac{\partial \tilde{\phi}(x_k)}{\partial x_{i,k}}\right|_{x_k=0} + \frac{1}{2!}\sum_{i=1}^{n_x}\sum_{j=1}^{n_x} x_{i,k}x_{j,k} \left.\frac{\partial^2 \tilde{\phi}(x_k)}{\partial x_{i,k}\,\partial x_{j,k}}\right|_{x_k=0} + \frac{1}{3!}\sum_{i=1}^{n_x}\sum_{j=1}^{n_x}\sum_{l=1}^{n_x} x_{i,k}x_{j,k}x_{l,k} \left.\frac{\partial^3 \tilde{\phi}(x_k)}{\partial x_{i,k}\,\partial x_{j,k}\,\partial x_{l,k}}\right|_{x_k=0}, \tag{53}$$

where $x_{i,k}$ is the i-th element of the vector x_k. The above equation can be rewritten as

$$\begin{aligned}\tilde{\phi}(x_k) &\approx \tilde{\phi}(0) + \sum_{i=1}^{n_x} x_{i,k} \left.\frac{\partial \tilde{\phi}(x_k)}{\partial x_{i,k}}\right|_{x_k=0} + \frac{1}{2}\sum_{i=1}^{n_x} x_{i,k}^2 \left.\frac{\partial^2 \tilde{\phi}(x_k)}{\partial x_{i,k}^2}\right|_{x_k=0} \\ &+ \sum_{i=1}^{n_x}\sum_{\substack{j=1\\ j\neq i}}^{n_x} x_{i,k}x_{j,k} \left.\frac{\partial^2 \tilde{\phi}(x_k)}{\partial x_{i,k}\partial x_{j,k}}\right|_{x_k=0} + \frac{1}{6}\sum_{i=1}^{n_x} x_{i,k}^3 \left.\frac{\partial^3 \tilde{\phi}(x_k)}{\partial x_{i,k}^3}\right|_{x_k=0} + \frac{1}{2}\sum_{i=1}^{n_x}\sum_{\substack{j=1\\ j\neq i}}^{n_x} x_{i,k}^2 x_{j,k} \left.\frac{\partial^3 \tilde{\phi}(x_k)}{\partial x_{i,k}^2\partial x_{j,k}}\right|_{x_k=0} \\ &+ \frac{1}{6}\sum_{i=1}^{n_x}\sum_{\substack{j=1\\ j\neq i}}^{n_x}\sum_{\substack{l=1\\ l\neq i, l\neq j}}^{n_x} x_{i,k}x_{j,k}x_{l,k} \left.\frac{\partial^3 \tilde{\phi}(x_k)}{\partial x_{i,k}\partial x_{j,k}\partial x_{l,k}}\right|_{x_k=0}. \end{aligned} \tag{54}$$

We use the following formula [35, pp. 38–39] to evaluate the value of A_k and B_k:

$$\int_{-\infty}^{\infty} x_i^n \mathcal{N}(x_i; 0, 1)dx = \begin{cases} 1 & n = 0, \\ 1 \times 3 \times 5 \times (n-1) & n \text{ even}, \\ 0 & n \text{ odd}. \end{cases} \tag{55}$$

From Eq. (12), A_k can be calculated as

$$\begin{aligned} A_k &= \int_{-\infty}^{\infty} \tilde{\phi}(x_k)\mathbf{H_1}(x_k)^T w(x_k)dx_k \\ &= \int_{-\infty}^{\infty} \tilde{\phi}(x_k)x_k^T \mathcal{N}(x_k; 0, I)dx_k. \end{aligned} \tag{56}$$

Substituting Eq. (54) and Eq. (55) in the above equation, we obtain Eq. (27). Similarly, we can calculate

$$B_k = \int_{-\infty}^{\infty} \tilde{\phi}(x_k)w(x_k)dx_k = \int_{-\infty}^{\infty} \tilde{\phi}(x_k)\mathcal{N}(x_k; 0, I)dx_k. \tag{57}$$

Substituting Eq. (54), the above equation yields Eq. (28).

APPENDIX B

Proof of Proposition 2: At $x_k = 0$, $\mathcal{X}_k = \hat{\mathcal{X}}_{k|k}$, and we use the nabla operator (∇), as defined above to write:

$$\begin{aligned} \sum_{i=1}^{n_x} \left.\frac{\partial \tilde{\phi}(x_k)}{\partial x_{i,k}}\right|_{x_k=0} e_i^T &= [\nabla_{x_k}\tilde{\phi}(x_k)^T]^T|_{x_k=0} \\ &= [\nabla_{x_k}\phi(\mathcal{X}_k)^T]^T|_{\mathcal{X}_k=\hat{\mathcal{X}}_{k|k}}. \end{aligned} \tag{58}$$

Using the vector chain rule [35, p. 103], we can write

$$\begin{aligned} \nabla_{x_k} &= [\nabla_{x_k}\mathcal{X}_k^T]\nabla_{\mathcal{X}_k} = \nabla_{x_k}[\hat{\mathcal{X}}_{k|k} + S_{k|k}x_k]^T\nabla_{\mathcal{X}_k} \\ &= S_{k|k}^T\nabla_{\mathcal{X}_k}. \end{aligned} \tag{59}$$

Substituting the above in Eq. (58),

$$\begin{aligned} \sum_{i=1}^{n_x} \left.\frac{\partial \tilde{\phi}(x_k)}{\partial x_{i,k}}\right|_{x_k=0} e_i^T &= [S_{k|k}^T\nabla_{\mathcal{X}_k}\phi(\mathcal{X}_k)^T]^T|_{\mathcal{X}_k=\hat{\mathcal{X}}_{k|k}} \\ &= [\nabla_{\mathcal{X}_k}\phi(\mathcal{X}_k)^T]^T|_{\mathcal{X}_k=\hat{\mathcal{X}}_{k|k}}S_{k|k}. \end{aligned} \tag{60}$$

The third order partial derivatives can be expressed with the help of ∇ operator as

$$\sum_{i=1}^{n_x}\sum_{j=1}^{n_x} \frac{\partial^3}{\partial x_{i,k}^2\partial x_{j,k}} e_j^T = \sum_{i=1}^{n_x}\sum_{j=1}^{n_x} M(i,j)e_i^T, \tag{61}$$

where

$$\begin{aligned} M &= \nabla_{x_k}\nabla_{x_k}^T\text{diag}(\nabla_{x_k}) \\ &= S_{k|k}^T\nabla_{\mathcal{X}_k}\nabla_{\mathcal{X}_k}^T S_{k|k}\text{diag}(S_{k|k}^T\nabla_{\mathcal{X}_k}). \end{aligned} \tag{62}$$

Substituting Eqs. (60) – (62) in Eq. (27), we receive

$$A_k = [\nabla_{\mathcal{X}_k}\phi(\mathcal{X}_k)^T]^T\Big|_{\mathcal{X}_k=\hat{\mathcal{X}}_{k|k}} S_{k|k} + \frac{1}{2}\sum_{i=1}^{n_x}\sum_{j=1}^{n_x} M(i,j)\,\phi(\mathcal{X}_k)|_{\mathcal{X}_k=\hat{\mathcal{X}}_{k|k}} e_i^T. \tag{63}$$

The second order partial derivatives can be expressed with the help of ∇ operator as

$$\sum_{i=1}^{n_x} \frac{\partial^2}{\partial x_{i,k}^2} = \text{trace}\{\nabla_{x_k}\nabla_{x_k}^T\} = \text{trace}\{S_{k|k}^T\nabla_{\mathcal{X}_k}\nabla_{\mathcal{X}_k}^T S_{k|k}\}. \tag{64}$$

Using the cyclic property of the trace *i.e.* under circular permutation trace is invariant, the above equation becomes

$$\begin{aligned} \sum_{i=1}^{n_x} \frac{\partial^2}{\partial x_{i,k}^2} &= \text{trace}\{S_{k|k}S_{k|k}^T\nabla_{\mathcal{X}_k}\nabla_{\mathcal{X}_k}^T\} \\ &= \text{trace}\{P_{k|k}\nabla_{\mathcal{X}_k}\nabla_{\mathcal{X}_k}^T\}. \end{aligned} \tag{65}$$

Substituting the above in Eq. (28), we receive Eq. (30).

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2004.
[3] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Chelmsford, MA, USA: Courier Corporation, 2012.
[4] T. Karvonen, "Stability of linear and non-linear Kalman filters," M.S. thesis, Dept. Math. Statist., Univ. Helsinki, Helsinki, Finland, 2014.
[5] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, "Stochastic stability of the discrete-time extended Kalman filter," *IEEE Trans. Autom. Control*, vol. 44, no. 4, pp. 714–728, Apr. 1999.
[6] 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.
[7] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Oct. 2000, pp. 153–158.

[8] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 910–927, May 2000.
[9] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.
[10] S. Bhaumik and Swati, "Cubature quadrature Kalman filter," *IET Signal Process.*, vol. 7, no. 7, pp. 533–541, Sep. 2013.
[11] B. Jia, M. Xin, and Y. Cheng, "High-degree cubature Kalman filter," *Automatica*, vol. 49, no. 2, pp. 510–518, Feb. 2013.
[12] J. Wang, T. Zhang, X. Xu, and Y. Li, "A variational Bayesian based strong tracking interpolatory cubature Kalman filter for maneuvering target tracking," *IEEE Access*, vol. 6, pp. 52544–52560, 2018.
[13] S. Bhaumik and P. Date, *Nonlinear Estimation: Methods and Applications With Deterministic Sample Points*. Boca Raton, FL, USA: CRC Press, 2019.
[14] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[15] G. Evensen, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean Dyn.*, vol. 53, no. 4, pp. 343–367, Nov. 2003.
[16] J. Duník, O. Straka, and M. Šimandl, "Stochastic integration filter," *IEEE Trans. Autom. Control*, vol. 58, no. 6, pp. 1561–1566, Jun. 2013.
[17] E. Blasch, J. Duník, O. Straka, and M. Šimandl, "Comparison of stochastic integration filter with the unscented Kalman filter for maneuvering targets," in *Proc. IEEE Nat. Aerosp. Electron. Conf. (NAECON)*, Jun. 2014, pp. 135–142.
[18] I. Arasaratnam and S. Haykin, "Square-root quadrature Kalman filtering," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2589–2593, Jun. 2008.
[19] S. Bhaumik and Swati, "Square-root cubature-quadrature Kalman filter," *Asian J. Control*, vol. 16, no. 2, pp. 617–622, Mar. 2014.
[20] I. Sarkas, D. Mavridis, M. Papamichail, and G. Papadopoulos, "Volterra analysis using Chebyshev series," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 1931–1934.
[21] E. Süli and D. F. Mayers, *An Introduction to Numerical Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[22] F. Buet-Golfouse, "A multinomial theorem for Hermite polynomials and financial applications," *Appl. Math.*, vol. 6, no. 6, pp. 1017–1030, 2015.
[23] M. Abramowitz and I. A. Stegun, "Handbook of mathematical functions with formulas, graphs, and mathematical tables," in *US Department of Commerce* (National Bureau of Standards Applied Mathematics Series 55). Washington, DC, USA: U.S. Government Printing Office, 1965.
[24] B. Holmquist, "The d-variate vector Hermite polynomial of order k," *Linear Algebra Appl.*, vols. 237–238, pp. 155–190, Apr. 1996.
[25] J. Sarmavuori and S. Särkkä, "Fourier-Hermite Kalman filter," *IEEE Trans. Autom. Control*, vol. 57, no. 6, pp. 1511–1515, Jun. 2012.
[26] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
[27] Y. Xu, L. Mili, and J. Zhao, "A novel polynomial-chaos-based Kalman filter," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 9–13, Jan. 2019.
[28] A. Germani, C. Manes, and P. Palumbo, "Polynomial extended Kalman filter," *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 2059–2064, Dec. 2005.
[29] M. B. Luca, S. Azou, G. Burel, and A. Serbanescu, "On exact Kalman filtering of polynomial systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 6, pp. 1329–1340, Jun. 2006.
[30] R. Karlsson, T. Schon, and F. Gustafsson, "Complexity analysis of the marginalized particle filter," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4408–4411, Nov. 2005.
[31] X. R. Li, Z. Zhao, and V. P. Jilkov, "Practical measures and test for credibility of an estimator," in *Proc. Workshop Estimation, Tracking, Fusion A, Tribute to Yaakov Bar-Shalom*, 2001, pp. 481–495.
[32] P. H. Leong, S. Arulampalam, T. A. 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.
[33] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell, "Comparison of EKF, pseudomeasurement, and particle filters for a bearing-only target tracking problem," in *Proc. Signal Data Process. Small Targets*, vol. 4728. Orlando, FL, USA: International Society for Optics and Photonics, Aug. 2002, pp. 240–250.
[34] R. Radhakrishnan, S. Bhaumik, and N. K. Tomar, "Gaussian sum shifted Rayleigh filter for underwater bearings-only target tracking problems," *IEEE J. Ocean. Eng.*, vol. 44, no. 2, pp. 492–501, Apr. 2019.
[35] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*. Hoboken, NJ, USA: Wiley, 2012.

KUNDAN KUMAR (Graduate Student Member, IEEE) received the B.Tech. degree in electrical engineering from the Muzaffarpur Institute of Technology, Muzaffarpur, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Indian Institute of Technology Patna. His research interests include state estimation and nonlinear filtering.

SHOVAN BHAUMIK received the B.Sc. degree in physics from the University of Calcutta, in 1999, and the B.Tech. degree in electronics and instrumentation engineering and the M.E. and Ph.D. degrees in electrical engineering from Jadavpur University in 2002, 2004, and 2009, respectively.

From 2007 to 2009, he was a Research Engineer with the GE Global Research, GE India Technology Center, Bangalore, India. He is currently working as an Associate Professor with the Department of Electrical Engineering, Indian Institute of Technology Patna. His research activities are supported by the Naval Research Board, Department of Science and Technology, Ministry of Electronics and Information Technology, Defence Research and Development Organization, India. He has published more than 25 refereed journal articles and supervised four Ph.D. students to completion, as the Principal Supervisor. He has authored a book *Nonlinear Estimation Methods and Applications with Deterministic Sample Points* (CRC press). His main research interests include nonlinear estimation, statistical signal processing, and underwater target tracking. He also received the Young Faculty Research Fellowship Award from the Ministry of Electronics and Information Technology, Government of India.

PARESH DATE received the master's degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, in 1995.

From 1995 to 1996, he has worked as an Engineering Executive with the Control and Automation Division, Larsen and Toubro Ltd., before starting his Ph.D. at the Cambridge University Engineering Department (CUED), U.K. In July 2002, he has worked with CUED, as a Research Associate, before joining Brunel University London. He is currently a Reader and the Director of Research with the Department of Mathematics. His research has been funded by Grants from the Engineering and Physical Sciences Research Council, U.K., from charitable bodies, such as the London Mathematical Society, the Royal Society, and the industry. He has held several short term visiting appointments with the Indian Institute of Technology, Mumbai, from 2006 to 2007, the Indian Institute of Management Kolkata, in 2010, the Indian Institute of Technology Patna, in 2013, 2014, 2016, and 2018, and a six month visiting professorship with the Indian Institute of Technology Gandhinagar, in 2019. He has published more than 50 refereed papers and supervised ten Ph.D. students to completion, as the Principal Supervisor. His principal research interests include altering and its applications, especially in financial mathematics. He is a Fellow of the Institute of Mathematics and its Applications and an Associate Editor for the *IMA Journal of Management Mathematics*.