


ORIGINAL RESEARCH

Polynomial chaos Kalman filter for target tracking applications

 Kundan Kumar¹  | Ranjeet Kumar Tiwari¹ | Shovan Bhaumik¹ | Paresh Date²
¹Department of Electrical Engineering, Indian Institute of Technology Patna, Patna, India

²Department of Mathematics, College of Engineering, Design and Physical Sciences, Brunel University London, London, UK

Correspondence

 Paresh Date, Department of Mathematics, College of Engineering, Design and Physical Sciences, Brunel University London, London, UK.
Email: paresh.date@brunel.ac.uk

Abstract

In this paper, an approximate Gaussian state estimator is developed based on generalised polynomial chaos expansion for target tracking applications. Motivated by the fact that calculating conditional moments in an approximate Gaussian filter involves computing integrals with respect to Gaussian density, the authors approximate the non-linear dynamics using polynomial chaos expansion. Second-order as well as third-order polynomial chaos expansions were used for approximate filtering, to derive the necessary recursive algorithm and also provide certain algebraic simplifications which reduce the computational burden without significantly affecting the filtering performance. Two comprehensive numerical experiments for multivariate systems, including one for a multi-model system, demonstrate the potential of the new algorithms.

KEYWORDS

bearings-only tracking, collocation points, Kalman filter, multiple models, polynomial chaos expansion, state estimation, target tracking

1 | INTRODUCTION

Target tracking [1–4] is a state estimation problem to estimate the kinematics of a target, such as the position, the velocity and the turn rate, from noisy measurements received from the sensors. In target tracking problems, the process and measurement equations are usually non-linear, and a closed-form optimal estimate is not available. There is no single non-linear filter which clearly outperforms all other filtering algorithms for a range of tracking problems. Hence, the selection of filtering algorithm depends on a trade-off between estimation accuracy and computational complexity, which is specific to each application. Several filtering heuristics have been developed over time to address this. The first popular non-linear filter was the extended Kalman filter (EKF) [5, 6]. It is based on first-order Taylor series approximation of the process and measurement functions around the previous estimate and uses the standard recursive Kalman filter (KF) structure. However, for highly non-linear systems, the EKF performs poorly [5, 7].

Due to the limitations of the EKF, more advanced estimation techniques have been developed. In the literature, mainly two approaches are available for non-linear filtering. In the first one, the prior and the posterior probability density

functions (PDFs) are approximated as Gaussian. Their mean and covariances are evaluated with a few deterministic sample points and the associated weights [8]. The cubature Kalman filter (CKF) [9], the interpolatory CKF (ICKF) [10], the embedded CKF (ECKF) [11], the unscented Kalman filter (UKF) [12, 13] and the Gauss-Hermite filter (GHF) [14], among others, are examples of such estimators. In another approach, the prior and posterior pdfs are approximated with a large number of random particles generated using importance sampling and their corresponding probability weights. It is popularly known as particle filter (PF) [15]. Although the PF and its variants can achieve high estimation accuracy, their computation demand is relatively higher than that of the first class of filters. This problem becomes worse with an increase in the system dimension.

In this paper, we will employ polynomial chaos to arrive at Gaussian prior and posterior densities of the state estimate. Polynomial chaos theory was first introduced in Ref. [16], where Hermite polynomial was used to model a stochastic process with the help of Gaussian random variables. Later, the authors in Ref. [17] presented generalised chaos theory by using orthogonal polynomials for different continuous and discrete distributions. In Ref. [18], the chaos expansion is used to represent power flow response. The chaos expansion is

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *IET Radar, Sonar & Navigation* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

reported to converge in \mathcal{L}_2 space for a general class of stochastic processes having finite second moments [17, 19]. Polynomial chaos-based square-root Kalman filter was developed in Ref. [20], where the coefficients are calculated from the ensemble Kalman filter and the filter gain is calculated from the coefficients. A few attempts have been made to develop an estimation method with the help of generalised polynomial chaos for a non-linear system adopting the EKF approach [21] and the ensemble Kalman filter approach [22]. Both the papers considered continuous stochastic processes rather than discrete time processes, which are (arguably) more realistic in target tracking applications with discretely sampled measurements. Further, for a highly non-linear system, the generalised polynomial chaos EKF (gPC-EKF) performs poorly [23].

A second-order polynomial chaos expansion of a multivariate non-linear function leads to terms of the form $x_i x_j$. In addition to these terms, a third-order polynomial chaos expansion leads to terms of the form $x_i x_j x_k$. We call the terms in the expansion with $i = j$ for second-order terms and $i = j = k$ for third-order terms as ‘alike terms’ and any terms with $i \neq j$ are called as ‘cross terms’. Since approximate Gaussian filters, such as the CKF, ICKF, ECKF and UKF, among others, involve computing conditional moments of the state with respect to Gaussian density, it is intuitively attractive to use a polynomial expansion of the underlying non-linear functions using Hermite polynomials (i.e. polynomial chaos expansion), which provide an optimal approximation in \mathcal{L}_2 sense. Recently, Xu et al. [23] used polynomial chaos expansion to approximate non-linear functions and subsequently derive an algorithm for estimating states of a discrete non-linear system. They considered only a second-order polynomial chaos expansion with alike terms. Further, they used noise covariances to distribute the collocation points (CPs) around the mean. This makes the algorithm sensitive with filter initialisation and estimation diverges in tracking the truth in many scenarios.

In this paper, we also adopt the polynomial chaos expansion to approximate a non-linear function. Our specific contributions to Ref. [23] are as follows: (i) Ref. [23] considered only second-order polynomial chaos expansion with alike terms. In this paper, we develop an estimation algorithm using up to third-order polynomial chaos expansion. (ii) Further, we consider both alike terms and cross terms in both second-order and third-order expansion, and evaluate the effect of neglecting cross terms both empirically through comprehensive examples and in terms of flop counts. (iii) Ref. [23] used $(2n + 1)$ number of CPs on the axis at a distance of $\sqrt{3}$ from the origin. Instead of using such highly constrained CPs, we use generalised CPs, for example, as suggested in Refs. [24, 25]. We propose a structured procedure to choose CPs for polynomial chaos filtering and show that the proposed filters with generalised CPs achieve a high degree of accuracy and low track loss compared to existing approximate Gaussian filters.

The developed filters are applied to two target tracking problems, where we have seen that the estimator derived in Ref. [23] provides a comparable results with the existing Gaussian filters such as the CKF [9], the ICKF [10] and the UKF [12, 13]. However, the proposed method provides more

accurate tracking compared to Ref. [23] and other existing Gaussian filters at a modest increment in computational cost. The computational budget required for the proposed filter is studied in terms of flop counts and is compared with the existing filters.

The rest of the paper is organised as follows. Section 2 presents the Gaussian approximated filter under the Bayesian framework. In Section 3, we first approximate the function using polynomial chaos expansion, and subsequently derive the polynomial chaos Kalman filter (PCKF). The computation complexity of the proposed PCKF is compared with the CKF and UKF in Section 4. The simulation results are presented in Section 5, followed by a brief discussion and conclusions.

2 | BAYESIAN FRAMEWORK OF APPROXIMATE GAUSSIAN FILTERING

The process and the measurement equation in discrete-time state space model are given as

$$\mathcal{X}_k = f(\mathcal{X}_{k-1}) + \eta_{k-1}, \quad (1)$$

$$\mathcal{Y}_k = b(\mathcal{X}_k) + \nu_k, \quad (2)$$

where $\mathcal{X}_k \in \mathbb{R}^n$ and $\mathcal{Y}_k \in \mathbb{R}^{n_y}$ are the state and measurement vectors. $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $b(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{n_y}$ are process and measurement function, respectively. The process noise, η_{k-1} , and measurement noise, ν_k , are assumed to be Gaussian with mean zero and covariance matrices Q_{k-1} and R_k , respectively. It is also assumed that the initial state, \mathcal{X}_0 , the process and measurement noise are independent to each other.

In Bayesian filtering, the state \mathcal{X}_k is being estimated recursively using the measurement $\mathcal{Y}_{1:k}$ in two steps: (i) prediction step (ii) update step. In the prediction step, using Chapman–Kolmogorov equation we construct the prior pdf $p(\mathcal{X}_k | \mathcal{Y}_{1:k-1})$ from the knowledge of $p(\mathcal{X}_{k-1} | \mathcal{Y}_{1:k-1})$:

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

In the update step, Bayes' rule is used to construct the posterior density function $p(\mathcal{X}_k | \mathcal{Y}_{1:k})$

$$p(\mathcal{X}_k | \mathcal{Y}_{1:k}) = \frac{p(\mathcal{Y}_k | \mathcal{X}_k) p(\mathcal{X}_k | \mathcal{Y}_{1:k-1})}{\int p(\mathcal{Y}_k | \mathcal{X}_k) p(\mathcal{X}_k | \mathcal{Y}_{1:k-1}) d\mathcal{X}_k}. \quad (4)$$

For a linear system, the prior and posterior density functions (Equations 3–4) remain Gaussian, and the first two moments of the posterior density can be obtained recursively using the Kalman filter. For a non-linear system, the prior and posterior pdfs are no longer Gaussian. However, many approximate Gaussian filtering algorithms assume them to be Gaussian and calculate the mean vector and the covariance matrix recursively [6, Chap. 6, 9, 12, 14]. The prior mean can be calculated as [9, 14, 26]

$$\hat{\mathcal{X}}_{k|k-1} = \int_{-\infty}^{\infty} f(\mathcal{X}_{k-1}) \mathcal{N}(\mathcal{X}_{k-1}; \hat{\mathcal{X}}_{k-1|k-1}, P_{k-1|k-1}) d\mathcal{X}_{k-1}. \quad (5)$$

The prior error covariance ($P_{k|k-1}^{\mathcal{X}\mathcal{X}}$), the expected value of measurement ($\hat{\mathcal{Y}}_{k|k-1}$), the covariance of measurement ($P_{k|k-1}^{\mathcal{Y}\mathcal{Y}}$) and the cross-covariance of state and measurement ($P_{k|k-1}^{\mathcal{X}\mathcal{Y}}$) can also be expressed in a similar way [9, 14]. For exact expressions, see Section 2 of Ref. [26].

After receiving the new measurement, \mathcal{Y}_k , we construct the posterior mean and the posterior covariance as

$$\hat{\mathcal{X}}_{k|k} = \hat{\mathcal{X}}_{k|k-1} + K_k (\mathcal{Y}_k - \hat{\mathcal{Y}}_{k|k-1}), \quad (6)$$

and

$$P_{k|k}^{\mathcal{X}\mathcal{X}} = P_{k|k-1}^{\mathcal{X}\mathcal{X}} - K_k P_{k|k-1}^{\mathcal{Y}\mathcal{Y}} K_k^T, \quad (7)$$

where the Kalman gain (K_k) is expressed as

$$K_k = P_{k|k-1}^{\mathcal{X}\mathcal{Y}} \left(P_{k|k-1}^{\mathcal{Y}\mathcal{Y}} \right)^{-1}. \quad (8)$$

In general, for any arbitrary non-linear function, the integrals in estimating the conditional mean and the conditional covariance matrices are computationally intractable. In this paper, we use polynomial chaos expansion of the non-linear function to evaluate these integrals.

3 | POLYNOMIAL CHAOS-BASED KALMAN FILTER

The polynomial chaos expansion is a series expansion of Gaussian random variables in terms of Hermite polynomials [27, p. 44]. In the next subsection, we discuss the functional approximation using polynomial chaos expansion and develop the filtering algorithm in subsequent subsections.

3.1 | Approximation using polynomial chaos expansion

Here, we approximate the transition function, $f(\mathcal{X})$, where the state \mathcal{X} follows Gaussian distribution with mean $\hat{\mathcal{X}}$ and covariance P . It is important to note that the approximation is not generally on the mean value; rather, it is determined by the sample points. This point is made clearer later in the discussion, when we talk about the prior mean. At first, we transform \mathcal{X} into a standard Gaussian random variable, \mathbf{x} , following $\mathcal{X} = \hat{\mathcal{X}} + S\mathbf{x}$, where $P = SS^T$. After the transformation, the function $f(\mathcal{X})$ becomes $f(\hat{\mathcal{X}} + S\mathbf{x})$, and its d th order chaos expansion can be written as [18, 28]

$$\begin{aligned} f(\hat{\mathcal{X}} + S\mathbf{x}) &\approx a_0 + \sum_{i_1=1}^n a_{i_1} H_1(x_{i_1}) \\ &+ \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} a_{i_1 i_2} H_2(x_{i_1}, x_{i_2}) \\ &+ \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} H_3(x_{i_1}, x_{i_2}, x_{i_3}) + \dots \\ &+ \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} \dots \sum_{i_d=1}^{i_{d-1}} a_{i_1 i_2 \dots i_d} H_d(x_{i_1}, x_{i_2}, \dots, x_{i_d}), \end{aligned} \quad (9)$$

where $a_{i_1 i_2 \dots i_d}$ are the coefficients of the chaos expansion, $[x_1 \ x_2 \ \dots \ x_n]^T = \mathbf{x}$, and $H_d(x_1, x_2, \dots, x_n)$ is the Hermite polynomial of degree d , which is given by [17, 18]

$$\begin{aligned} H_d(x_1, x_2, \dots, x_n) &= (-1)^d \exp\left(\frac{1}{2}\mathbf{x}^T \mathbf{x}\right) \frac{\partial^n}{\partial x_1 \partial x_2 \dots \partial x_n} \\ &\exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{x}\right). \end{aligned} \quad (10)$$

Hermite polynomials are orthogonal in the inner product sense [27, p. 45] that is, $E[H_p H_q] = 0$ if $H_p \neq H_q$, where the expectation is taken with respect to the standard Gaussian measure. Further, polynomial chaos expansion as expressed in Equation (9) converges in the mean square sense to the true function [27, 28] and the squared error of approximation decreases with the increase in order of expansion. In Refs. [18, 27, 28], it has been mentioned that the approximation of degree more than three provides only a marginal improvement in accuracy and is overshadowed by the increase in computational burden.

In this work, we confine ourselves to chaos expansion up to third order while approximating the non-linear state transition function and the measurement function. From Equation (9), we approximate the transition function with chaos expansion of third order as follows:

$$\begin{aligned} f(\hat{\mathcal{X}} + S\mathbf{x}) &\approx a_0 + \sum_{i_1=1}^n a_{i_1} H_1(x_{i_1}) + \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} a_{i_1 i_2} H_2(x_{i_1}, x_{i_2}) \\ &+ \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} H_3(x_{i_1}, x_{i_2}, x_{i_3}) \\ &\approx a_0 + AH(\mathbf{x}), \end{aligned} \quad (11)$$

where the matrix $H(\mathbf{x})$ and A are given by

$$H(\mathbf{x})_{(m-1) \times 1} = [H_1(x_1) \ \dots \ H_3(x_n, x_n, x_n)]^T, \quad (12)$$

and

$$A_{n \times (m-1)} = [a_1 \ a_2 \ \dots \ a_{nnn}], \quad (13)$$

respectively. The total number of terms in the approximated polynomial is denoted by m , and for a third-order polynomial chaos expansion, $m = \binom{n+3}{3}$. Note that as per our nomenclature in the Section 1, a_0 and coefficient a with subscript $i_1 = i_2$ for second-order terms and subscript $i_1 = i_2 = i_3$ for third-order terms in Equation (11) are alike terms and the rest of the coefficients are cross terms.

From Equation (10), we know that

$$H_1(x_{i_1}) = x_{i_1},$$

$$H_2(x_{i_1}, x_{i_1}) = \frac{(x_{i_1}^2 - 1)}{\sqrt{2}}, H_2(x_{i_1}, x_{i_2}) = x_{i_1}x_{i_2},$$

$$H_3(x_{i_1}, x_{i_1}, x_{i_1}) = \frac{(x_{i_1}^3 - 3x_{i_1})}{\sqrt{6}}, H_3(x_{i_1}, x_{i_2}, x_{i_2}) = \frac{x_{i_1}(x_{i_2}^2 - 1)}{\sqrt{2}},$$

$$H_3(x_{i_1}, x_{i_2}, x_{i_3}) = x_{i_1}x_{i_2}x_{i_3}.$$

Similar to Equation (11), we approximate the measurement function (Equation 2) as

$$\begin{aligned} h(\hat{\mathcal{X}} + Sx) &\approx b_0 + \sum_{i_1=1}^n b_{i_1}H_1(x_{i_1}) + \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} b_{i_1 i_2}H_2(x_{i_1}, x_{i_2}) \\ &\quad + \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} b_{i_1 i_2 i_3}H_3(x_{i_1}, x_{i_2}, x_{i_3}) \\ &\approx b_0 + BH(x), \end{aligned} \quad (14)$$

where B is a matrix of dimension $n_y \times (m - 1)$, and can be expressed in a similar fashion as Equation (13).

3.2 | Polynomial chaos approximation-based filter

In this subsection, we derive the filtering algorithm for the following approximate non-linear system:

$$\begin{aligned} \mathcal{X}_k &= a_0 + AH(\mathcal{X}_{k-1}) + \eta_{k-1}, \\ \mathcal{Y}_k &= b_0 + BH(\mathcal{X}_k) + \nu_k, \end{aligned} \quad (15)$$

where a_0 , b_0 , A and B are as defined in the previous section and $\mathcal{X}_k = \hat{\mathcal{X}}_k + Sx$, with S being a Cholesky factor of the covariance matrix of \mathcal{X}_k and x being a random variable with standard normal distribution. The following three lemmas allow computation of moments using polynomial chaos expansion of non-linear functions.

Remark 1 For a standard Gaussian random variable x , the matrix $H(x)$ has the following properties:

$$E[H(x)] = 0, \quad (16)$$

$$E\left[H(x)H(x)^T\right] = I. \quad (17)$$

Lemma 1 For the system in Equation (15), the prior mean and the error covariance are given by

$$\hat{\mathcal{X}}_{k|k-1} = a_0, \quad (18)$$

$$P_{k|k-1}^{\mathcal{X}\mathcal{X}} = AA^T + Q_{k-1}. \quad (19)$$

Proof The prior mean can be calculated as

$$\hat{\mathcal{X}}_{k|k-1} = E[\mathcal{X}_k | \mathcal{Y}_{1:k-1}].$$

Now, using the affine transformation [29, pp. 36–37] $\mathcal{X}_{k-1} = \hat{\mathcal{X}}_{k-1|k-1} + S_{k-1|k-1}x_{k-1}$, the above equation becomes

$$\begin{aligned} \hat{\mathcal{X}}_{k|k-1} &= E\left[f\left(\hat{\mathcal{X}}_{k-1|k-1} + S_{k-1|k-1}x_{k-1}\right) + \eta_{k-1}\right] \\ &= E[a_0 + AH(x_{k-1}) + \eta_{k-1}] \\ &= a_0. \end{aligned}$$

The predicted estimation error is

$$\mathcal{X}_k - \hat{\mathcal{X}}_{k|k-1} = AH(x_{k-1}) + \eta_{k-1}. \quad (20)$$

The prior error covariance can be calculated as

$$\begin{aligned} P_{k|k-1}^{\mathcal{X}\mathcal{X}} &= E\left[\left(\mathcal{X}_k - \hat{\mathcal{X}}_{k|k-1}\right)\left(\mathcal{X}_k - \hat{\mathcal{X}}_{k|k-1}\right)^T | \mathcal{Y}_{1:k-1}\right] \\ &= E\left[(AH(x_{k-1}) + \eta_{k-1})(AH(x_{k-1}) + \eta_{k-1})^T\right] \\ &= AE\left[H(x_{k-1})H(x_{k-1})^T\right]A^T + Q_{k-1} \\ &= AA^T + Q_{k-1}. \end{aligned}$$

Lemma 2 For the system in Equation (15), the expected value of one-step-ahead measurement, $\hat{\mathcal{Y}}_{k|k-1}$, the covariance of the same measurement, $P_{k|k-1}^{\mathcal{Y}\mathcal{Y}}$, and the cross covariance of the state and the measurement, $\left(P_{k|k-1}^{\mathcal{X}\mathcal{Y}}\right)$, are, respectively, given by

$$\hat{\mathcal{Y}}_{k|k-1} = b_0, \quad (21)$$

$$P_{k|k-1}^{\mathcal{Y}\mathcal{Y}} = BB^T + R_k, \quad (22)$$

$$P_{k|k-1}^{\mathcal{X}\mathcal{Y}} = AB^T. \quad (23)$$

Proof The expected value of measurement is

$$\widehat{\mathcal{Y}}_{k|k-1} = E[\mathcal{Y}_k | \mathcal{Y}_{1:k-1}].$$

Using Equations (2) and (14), the above equation can be written as

$$\widehat{\mathcal{Y}}_{k|k-1} = E[(b_0 + BH(x_k) + \nu_k) | \mathcal{Y}_{1:k-1}] = b_0.$$

Now, since

$$\mathcal{Y}_k - \widehat{\mathcal{Y}}_{k|k-1} = BH(x_k) + \nu_k, \quad (24)$$

the covariance of the measurement can be expressed as

$$\begin{aligned} P_{k|k-1}^{\mathcal{Y}\mathcal{Y}} &= E\left[(\mathcal{Y}_k - \widehat{\mathcal{Y}}_{k|k-1})(\mathcal{Y}_k - \widehat{\mathcal{Y}}_{k|k-1})^T | \mathcal{Y}_{1:k-1}\right] \\ &= E\left[(BH(x_k) + \nu_k)(BH(x_k) + \nu_k)^T\right] \\ &= BE\left[H(x_k)H(x_k)^T\right]B^T + R_k = BB^T + R_k. \end{aligned} \quad (25)$$

Further, the cross-covariance can be calculated as

$$\begin{aligned} P_{k|k-1}^{\mathcal{X}\mathcal{Y}} &= E\left[(\mathcal{X}_k - \widehat{\mathcal{X}}_{k|k-1})(\mathcal{Y}_k - \widehat{\mathcal{Y}}_{k|k-1})^T | \mathcal{Y}_{1:k-1}\right] \\ &= AB^T. \end{aligned} \quad (26)$$

3.3 | Determining the coefficients

Next, we discuss how to calculate the coefficients a_0 , b_0 , A and B . From Equation (11)

$$\begin{aligned} f(\widehat{\mathcal{X}} + Sx) &= a_0 + AH(x) \\ &= [a_0 \quad A]_{m \times m} \begin{bmatrix} 1 \\ H(x) \end{bmatrix}_{m \times 1} \\ &= A'H'(x). \end{aligned} \quad (27)$$

Our objective is to determine the matrix A' . As there are m coefficients to be evaluated, we need m independent linear equations. We generate these equations using m CPs; $\xi_i \in \mathbb{R}^n$, $i = \{1, 2, \dots, m\}$. This gives mn equations such that

$$\chi = \mathbf{H}A'^T, \quad (28)$$

where χ is a matrix with dimension $m \times n$ given by

$$\chi = \begin{bmatrix} f^T(\widehat{\mathcal{X}} + S\xi_1) \\ f^T(\widehat{\mathcal{X}} + S\xi_2) \\ \vdots \\ f^T(\widehat{\mathcal{X}} + S\xi_m) \end{bmatrix}_{m \times n},$$

and the matrix $\mathbf{H} \in \mathbb{R}^{m \times m}$ is given by

$$\mathbf{H} = \begin{bmatrix} H'^T(\xi_1) \\ H'^T(\xi_2) \\ \vdots \\ H'^T(\xi_m) \end{bmatrix}_{m \times m}.$$

Once we solve the above linear equation, we receive

$$A'^T = \mathbf{H}^{-1}\chi. \quad (29)$$

We can also evaluate the matrix $B' = [b_0 \quad B]$ in a similar fashion. Note that the prior mean $\widehat{\mathcal{X}}_{k|k-1} = a_0$, as given by Equation (18), is effectively determined by our choice of CPs.

3.4 | Generation of collocation points

As mentioned earlier, the CPs are a set of points in n dimensional real space which are required to calculate the coefficients of polynomial chaos expansion. As the accuracy of the proposed filter depends on the choice of CPs, it is advised to choose a set of CPs that can capture high probability density region [24, 25, 27]. For a normal distribution, the mean and the mode coincide, and the high probability region is concentrated around the mean. Firstly, single dimension CPs are generated by solving the roots of $(d+1)$ th Hermite polynomial. For example, if we want to proceed with the second-order polynomial chaos approximation, we need to find out the roots of a third-order Hermite polynomial. Next, these points are combined using the product rule to create a set of suitable CPs. If the origin is not included in the set of points, it is added. As mentioned above, for a standard normal random variable, the origin is the mode that is, the point of highest probability. Thus, the set of probable CPs consists of $(d+1)^n$ or $(d+1)^n + 1$ number of points for even or odd order of polynomial chaos approximation, respectively. From the set of acceptable CPs, we need to select m linearly independent points and solve the system of linear equations which results in the values of the coefficients. Now, the question is which m CPs to select from the set. For this, we prioritise the CPs which are near to the origin and check the rank condition till the desired number of support points are collected. Please see the Algorithm 1 for the generation of support points and see Algorithm 2 for the filtering algorithm (termed PCKF). It is worth mentioning

that the selection of CPs is not unique, and other choices are possible, including suggestions in Refs. [30–34]. Note that when we initialise the filter, $\hat{\mathcal{X}}_{0|0}$ is not the mean; it is a sample point drawn from a known prior distribution [5, p. 246, 6, p. 54]. Of course, the mean itself can be chosen as the sample point in question, although the effect of bias due to sampling of initial value tends to dissipate quite quickly for most practical systems.

Algorithm 1 Generation of CPs

- Calculate the roots of $(d + 1)$ th order Hermite polynomial
 - Use product rule to generate a set of probable CPs, $\mathbf{m}_{n \times (d+1)^n}$
 - if $(d + 1)$ is even
 - \mathbf{m} = concatenate (\mathbf{m} , $0_{n \times 1}$)
 - end if
 - Calculate the Euclidean distance of CPs, and ascend them to get $\xi = [\xi_1 \ \xi_2 \ \dots \ \xi_{(d+1)^n}]$
 - Calculate $\mathbf{H} = H^T(\xi_1)$
 - for $i = 2 : \text{length}(\xi(1, :))$
 - $\mathbf{H}\mathbf{H}$ = concatenate (\mathbf{H} , $H^T(\xi_i)$)
 - if $\mathbf{H}\mathbf{H}$ is full row rank
 - $\mathbf{H} = \mathbf{H}\mathbf{H}$
 - else
 - $i = i + 1$
 - end
 - if rank (\mathbf{H}) = m , break
 - end for
-

Remark 2 The selection of CPs includes some additional computation burden. However, this is independent of the filtering algorithm and the CPs can be computed in advance.

Remark 3 In some degenerate models, it might be the case that all points merge in some clusters such that it is not possible to define a full rank matrix $\mathbf{H}_{m \times m}$. In this case, the coefficients will be non-unique and a least squares solution of Equation (28) can still be used. In this paper, we have not accounted for such a case.

Remark 4 A mixed strategy can be adopted for time and measurement updates if the process equation is linear and the measurement equation is non-linear. As the process equation is linear, the time update step can be performed by the standard Kalman filtering algorithm, and the measurement update step is performed using the non-linear Gaussian filters.

Remark 5 If the process and measurement equations are non-linear, different orders of expansion can be used. Depending on the degree of linearity, each step (time update and measurement update) could be treated differently. Although it can be adopted, we have used the same order of expansion throughout the paper.

4 | COMPUTATIONAL COMPLEXITY

Here, in terms of floating-point operations (flops) count, we calculate the computational complexity of the proposed PCKF and compared it with the CKF and UKF. The addition or subtraction of any two matrix $A \in \mathbb{R}^{i \times j}$ and $B \in \mathbb{R}^{i \times j}$ requires ij flops. The multiplication of any two matrices A and B with dimension $i \times j$ and $j \times l$ requires $il(2j - 1)$ flops [35, 36]. The Cholesky and inverse operation of any square matrix $A \in \mathbb{R}^{i \times i}$ require $i^3/3 + 2i^2$ and i^3 flops, respectively [35, 36]. Total cumulative computational complexity of the PCKF algorithm in terms of flops is

$$C(n, n_y, m) = 2m^2(n + n_y) + m(6n^2 - n - n_y + 2n_y^2 + 2nn_y) + 2/3n^3 + 2n^2 - 2n_y^2 - 3nn_y + 4nn_y^2 + n_y^3 + n_y + 2n^2n_y, \quad (30)$$

where n , n_y and m are defined as before. The cumulative computation burden of the CKF and UKF with m numbers of sample points are given by

$$C(n, n_y, m) = m(6n^2 + 5n + 5n_y + 2n_y^2 + 2nn_y) + 2/3n^3 + 5n^2 + n_y^2 + 4nn_y^2 + n_y^3 + n_y + 2n^2n_y. \quad (31)$$

Remark 6 From the Algorithm 2, note that the inverse of the polynomial basis matrix, \mathbf{H} , can be calculated outside the estimation loop. Hence, flops count for computing \mathbf{H}^{-1} , which is $2m^3$, is not included in the Equation (30).

From the above two equations, it can be realized that the flops count for PCKFs are high, particularly so for a high dimensional system. To illustrate this, we plotted the number of flops with the dimension of state, n , keeping the ratio fixed as $n_y/n = 1/3$, in Figure 1. From the figure, it can be seen that the flops count of PCKF increases considerably with the dimension of the system. This is mainly due to a large number of cross terms for a large n . To reduce the computational burden without compromising the accuracy too much, we also experimented with polynomial chaos, truncated with only alike terms. Such truncated polynomial chaos filters with second- and third-order chaos expansion are represented with PCKF-2t and PCKF-3t, respectively. Polynomial chaos filters with second- and third-order expansion, with all the terms, are represented as PCKF-2 and PCKF-3, respectively. The proposed PCKF with second-order chaos expansion and alike terms of the third order is represented as PCKF-2,3t. Note that PCKF-2t is a modified version of the algorithm in Ref. [23], with generalised CPs. The total number of support points required for various PCKFs are tabulated in Table 1. Further, from Figure 1, we see that the flops counts for PCKF-2t and PCKF-3t reduce substantially when compared to PCKF-2 and PCKF-3, and they are comparable to those of the CKF and the UKF.

FIGURE 1 Flops count versus state dimension (n) plot of the different filters for $n_y/n = 1/3$, where n varies from 3 to 30

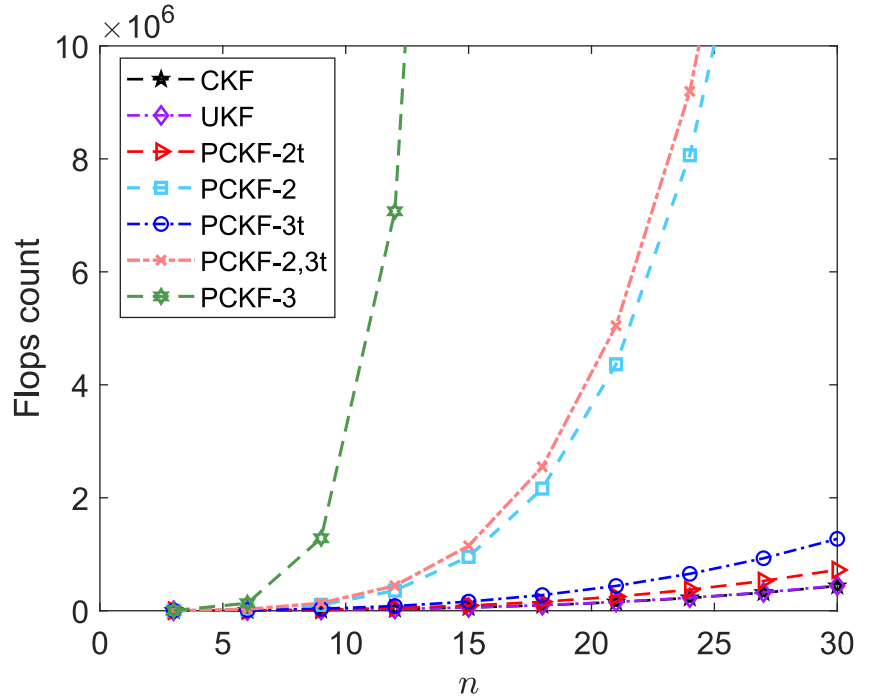


TABLE 1 No. of support points requirement for various PCKFs

Filter	PCKF-2t	PCKF-2	PCKF-3t	PCKF-2,3t	PCKF-3
M	$1 + 2n$	$\binom{n+2}{2}$	$1 + 3n$	$\binom{n+2}{2} + n$	$\binom{n+3}{3}$

Abbreviation: PCKF, polynomial chaos Kalman filter.

Algorithm 2 Pseudo code for polynomial chaos filter

Step 1: Initialisation

- Initialise the filter with $\hat{\mathcal{X}}_{0|0} \sim \mathcal{N}(\mathcal{X}_0, P_{0|0})$.
- Calculate collocation points (CPs), ξ , and polynomial chaos basis matrix \mathbf{H} .

Step 2: Time update

- Perform the Cholesky decomposition of $P_{k-1|k-1} \mathcal{X}_{k-1|k-1}^T$:

$$P_{k-1|k-1}^{\mathcal{X}\mathcal{X}} = S_{k-1|k-1} S_{k-1|k-1}^T.$$

- Evaluate the sample points around $\hat{\mathcal{X}}_{k-1|k-1}$:

$$\mathcal{X}_{i,k-1|k-1} = \hat{\mathcal{X}}_{k-1|k-1} + S_{k-1|k-1} \xi_i,$$

for $i = 1, 2, \dots, m$.

- Propagate the sample points:

$$\mathcal{X}_{i,k|k-1} = f(\mathcal{X}_{i,k-1|k-1}).$$

- Calculate the state prediction matrix,

$$\chi = [\mathcal{X}_{1,k|k-1}^T \quad \mathcal{X}_{2,k|k-1}^T \quad \dots \quad \mathcal{X}_{m,k|k-1}^T]^T.$$

- Calculate the coefficient matrix, $A' = (\mathbf{H}^{-1} \chi)^T$.

- Calculate $\hat{\mathcal{X}}_{k|k-1}$ and $P_{k|k-1}^{\mathcal{X}\mathcal{X}}$ using Equations (18) and (19), respectively.

Step 3: Measurement update

- Factorise $P_{k|k-1}^{\mathcal{X}\mathcal{X}} = S_{k|k-1} S_{k|k-1}^T$.
- Generate sample points around $\hat{\mathcal{X}}_{k|k-1}$:

$$\mathcal{X}_{i,k|k-1} = \hat{\mathcal{X}}_{k|k-1} + S_{k|k-1} \xi_i.$$

- Propagate the sample points with the measurement function

$$\mathcal{Y}_{i,k|k-1} = h(\mathcal{X}_{i,k|k-1}), \quad i = 1, 2, \dots, m.$$

- Calculate $\mathcal{Y} = [\mathcal{Y}_{1,k|k-1}^T \quad \mathcal{Y}_{2,k|k-1}^T \quad \dots \quad \mathcal{Y}_{m,k|k-1}^T]^T$.
- The coefficient matrix is calculated using $B^T = \mathbf{H}^{-1} \mathcal{Y}$.
- Evaluate $\hat{\mathcal{Y}}_{k|k-1}$, $P_{k|k-1}^{\mathcal{Y}\mathcal{Y}}$ and $P_{k|k-1}^{\mathcal{X}\mathcal{Y}}$ using Equations (21)–(23).
- Calculate $\hat{\mathcal{X}}_{k|k}$ and $P_{k|k}^{\mathcal{X}\mathcal{X}}$ using Equations (6) and (7), respectively.

5 | SIMULATION EXPERIMENTS

The developed filters are applied to the following two problems: (i) a manoeuvring aircraft represented with an interacting multiple model (IMM); (ii) multi-sensor bearings-only target tracking.

$$f_2(\mathcal{X}_k) = \begin{bmatrix} 1 & 0 & 0 & \sin(\omega_k T)/\omega_k & -(1 - \cos(\omega_k T))/\omega_k & 0 & 0 \\ 0 & 1 & 0 & (1 - \cos(\omega_k T))/\omega_k & \sin(\omega_k T)/\omega_k & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & \cos(\omega_k T) & -\sin(\omega_k T) & 0 & 0 \\ 0 & 0 & 0 & \sin(\omega_k T) & \cos(\omega_k T) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (34)$$

Problem 1 A manoeuvring aircraft's dynamics is represented with multiple models [37], in discrete time domain with the following equation:

$$\mathcal{X}_{k+1} = f_{\mu_k}(\mathcal{X}_k) + g_{\mu_k}(\mathcal{X}_k)\eta_{\mu_k},$$

where $\mu_k \in \{1, 2, 3\}$. The process is assumed to follow either constant velocity (CV), coordinated turn (CT) or constant acceleration (CA) model. $\mu_k = 1$ represents the CV model. Similarly, $\mu_k = 2$ and $\mu_k = 3$ correspond to the CT and the CA model, respectively. η_{μ_k} is the process noise for model μ_k following the Gaussian distribution with zero mean and identity covariance matrix. Note that the dimension of the states and state transition function $f_{\mu_k}(\cdot)$ are not fixed and they change with the selected model.

For CV model that is, $\mu_k = 1$, the state $\mathcal{X}_k = [x_k \ y_k \ z_k \ v_{x_k} \ v_{y_k}]^T$, where x_k and v_{x_k} represent position and velocity along x axis, respectively. The process and input functions for that model are

$$f_1(\mathcal{X}_k) = \begin{bmatrix} I_{2 \times 2} & 0_{2 \times 1} & T I_{2 \times 2} \\ 0_{2 \times 2} & I_{2 \times 2} & 0_{2 \times 1} \\ 0_{1 \times 2} & 0_{1 \times 2} & 1 \end{bmatrix} \mathcal{X}_k, \quad (32)$$

and

$$g_1(\mathcal{X}_k) = \alpha \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & 0 & \frac{1}{2}T^2 \\ T & 0 & 0 \\ 0 & T & 0 \end{bmatrix}, \quad (33)$$

respectively. α is the acceleration (also known as maneuverability coefficient). For CT model that is, $\mu_k = 2$, the state vector becomes $\mathcal{X}_k = [x_k \ y_k \ z_k \ v_{x_k} \ v_{y_k} \ v_{z_k} \ \omega_k]^T$, where ω_k is the turn rate and the process function in such a case become

The input function for CT model is expressed as

$$g_2(\mathcal{X}_k) = \begin{bmatrix} S_1 & 0_{3 \times 3} & 0_{3 \times 1} \\ 0_{3 \times 3} & S_1 & 0_{3 \times 1} \\ 0_{1 \times 3} & 0_{1 \times 3} & \frac{1}{\sqrt{v_{x_k}^2 + v_{y_k}^2 + v_{z_k}^2}} \end{bmatrix} C, \quad (35)$$

where

$$S_1 = \begin{bmatrix} \cos(\tan^{-1}(v_{y_k}/v_{x_k})) & -\sin(\tan^{-1}(v_{y_k}/v_{x_k})) & 0 \\ \sin(\tan^{-1}(v_{y_k}/v_{x_k})) & \cos(\tan^{-1}(v_{y_k}/v_{x_k})) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The matrix C is defined as

$$C = \begin{bmatrix} \frac{1}{2}\alpha_x T^2 & 0 & 0 \\ 0 & \frac{1}{2}\alpha_y T^2 & 0 \\ 0 & 0 & \frac{1}{2}\alpha_z T^2 \\ \alpha_x T & 0 & 0 \\ 0 & \alpha_y T & 0 \\ 0 & 0 & \alpha_z T \\ 0 & \alpha_y & 0 \end{bmatrix},$$

where α_x , α_y and α_z are the components of acceleration which are constant throughout the simulation. For CA model that is, $\mu_k = 3$, the state vector becomes $\mathcal{X}_k = [x_k \ y_k \ z_k \ v_{x_k} \ v_{y_k} \ v_{z_k} \ a_{z_k}]^T$, where a_{z_k} is the acceleration along z axis. The process and the input functions of such model are

$$f_3(\mathcal{X}_k) = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & T^2/2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (36)$$

and

$$g_3(\mathcal{X}_k) = \left[\left(\frac{1}{2} \alpha_z' T^2 I_{3 \times 3} \right)^T \quad (\alpha_z' T I_{3 \times 3})^T \quad [0 \quad 0 \quad \alpha_z']^T \right]^T, \quad (37)$$

respectively.

The measurements available to the estimator are the range, the bearing and the elevation angle of the target. We can write the measurement equation as

$$\mathcal{Y}_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \tan^{-1}(y_k/x_k) \\ \tan^{-1}(z_k/\sqrt{x_k^2 + y_k^2}) \end{bmatrix} + \nu_k, \quad (38)$$

The measurement noise, ν_k , is assumed to be Gaussian with mean zero and covariance, $R_k = \text{diag}([\sigma_r^2 \quad \sigma_b^2 \quad \sigma_a^2])$.

The true trajectory of the target shown in Figure 2 is constructed by the combination of following mode-switching scenarios: During the first six time-steps, the target follows the straight line, constant velocity trajectory (CV model). The next ten time-steps are hybrid between the CT and CA model, and the next eight time-steps are the hybrid between CV and CT model. The last four time-steps correspond to the CA

model. The sampling time T is assumed here 0.25 s, and total observation period lasted for 7 s. The initial truth of the target is $\mathcal{X}_0 = [5000 \text{ m } 5000 \text{ m } 1000 \text{ m } 300 \text{ m/s } 300 \text{ m/s } 100 \text{ m/s } -(\pi/60) \text{ rad } 5 \text{ m/s}^2]^T$, and the parameters used in this simulation work are listed in Table 2. The newly developed filters viz. PCKF-2, PCKF-2t, PCKF-2,3t, PCKF-3 and PCKF-3t were compared with some of the established non-linear filtering heuristics viz. the EKF, the CKF, the UKF and third degree ICKF (ICKF-3). Although several variants of the UKF are available in literature such as the scaled UKF [38], here we are comparing with the standard UKF, proposed in Refs. [12, 13]. We set the tuning parameter of the UKF, $\kappa = 3 - n$ as discussed in Ref. [12]. All the filters were initialised with the estimate, $\hat{\mathcal{X}}_{0|0} \sim \mathcal{N}(\mathcal{X}_0, P_{0|0})$, where $P_{0|0}$ is the initial error covariance given as $P_{0|0} = 10^3 \times \text{diag}([5 \ 5 \ 5 \ 4 \ 4 \ 4 \ 10^{-7} \ 2 \times 10^{-2}])$.

It is assumed that the mode-switching process is Markov and is given as

TABLE 2 Tracking parameters

Parameters	Values
A	0.5 m/s ²
α_x	15 m/s ²
α_y	20 m/s ²
α_z	15 m/s ²
α_z'	20 m/s ²
σ_r	20 m
σ_b	4×10^{-3} rad
σ_a	4×10^{-3} rad
q_2	5

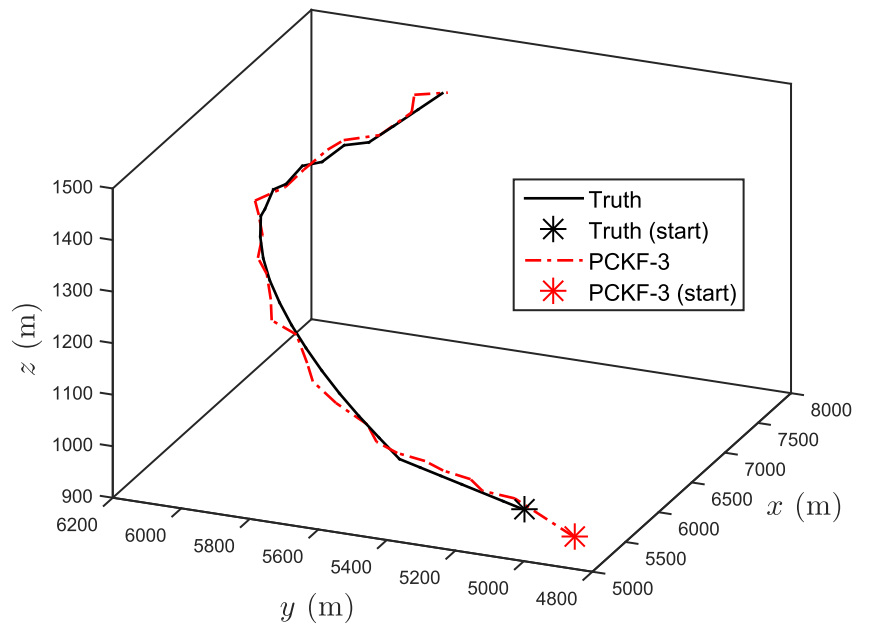


FIGURE 2 Truth target trajectory and estimated PCKF-3 for a single representative run. PCKF, polynomial chaos Kalman filter.

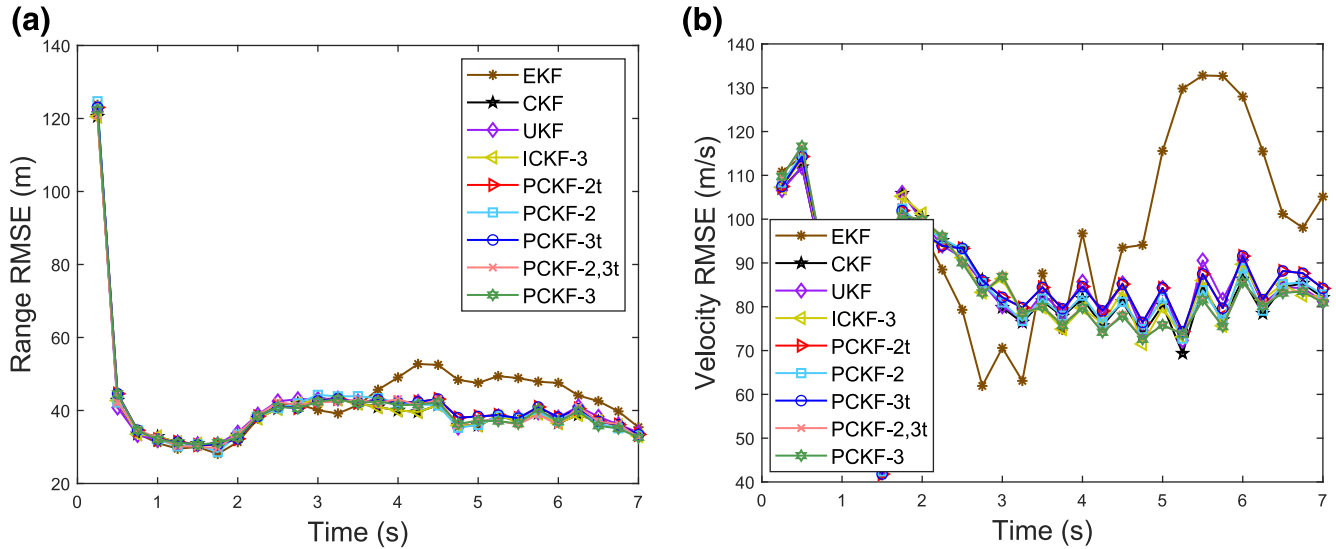


FIGURE 3 (a) Range RMSE; (b) Velocity RMSE of the different filters obtained from 500 MC runs (problem 1). CKF, cubature Kalman filter; EKF, extended Kalman filter; ICKF, interpolatory CKF; RMSE, root mean square error; UKF, unscented Kalman filter

$$p_{ij} = \begin{bmatrix} 0.90 & 0.05 & 0.05 \\ 0.05 & 0.90 & 0.05 \\ 0.05 & 0.05 & 0.90 \end{bmatrix}.$$

While this choice is somewhat arbitrary, it was found that the results are qualitatively very similar for diagonal elements of the Markov transition matrix between 0.8 and 0.98 [5, p. 457]. In a realistic simulation, the values would be chosen according to the switching frequency among the modes. A higher value of diagonal elements leads to a lower overall root mean square error (RMSE) and a higher peak RMSE during mode transition, whereas a lower value yields a larger overall RMSE and a lower peak RMSE during transition. For this problem, PCKF-3 successfully tracks the target, as illustrated in Figure 2.

The performance of all the filters is compared in terms of the RMSE, track loss and execution time. The RMSE in range at a time-step is defined as $\text{RMSE}_{\text{ran},k} = \left(\frac{1}{M} \sum_{i=1}^M (x_k^i - \hat{x}_k^i)^2 + (y_k^i - \hat{y}_k^i)^2 + (z_k^i - \hat{z}_k^i)^2 \right)^{1/2}$, where M is total number of ensembles, (x_k^i, y_k^i, z_k^i) and $(\hat{x}_k^i, \hat{y}_k^i, \hat{z}_k^i)$ are the true and estimated positions at the k th time-step of i th ensemble. Similar to the range RMSE, we may also calculate the resultant velocity RMSE. Again, an estimated target track for a single run is said to be diverging if the absolute error at the last step exceeds a predefined threshold (denoted by e_t). Figure 3a,b show the range and velocity RMSE for the different filters calculated over 500 MC runs (excluding the diverged tracks). Here, we choose the threshold value for track loss as $e_t = 50$ m. From the range RMSE plot (Figure 3a), it can be observed that the EKF attains the highest RMSE whereas other filters have comparable RMSE results. The velocity RMSE plot also exhibits the similar pattern. Here, it should be noted that the RMSE results of different filters (except the EKF) look similar because we have plotted RMSEs excluding the track divergence criteria. We also provide the mean of the RMSE value (average

TABLE 3 Average RMSE in range and velocity (problem 1)

Filter	Position (m)	Velocity (m/s)
EKF	44.65	93.65
CKF	40.90	83.20
UKF	41.16	83.60
ICKF-3	40.80	82.93
PCKF-2t	41.18	83.70
PCKF-2	40.84	83.06
PCKF-3t	40.87	83.50
PCKF-2,3t	40.79	82.32
PCKF-3	40.78	82.24

Abbreviations: CKF, cubature Kalman filter; EKF, extended Kalman filter; ICKF, interpolatory CKF; PCKF, polynomial chaos Kalman filter; RMSE, root mean square error; UKF, unscented Kalman filter.

RMSE) for position and velocity, obtained from 500 MC runs in Table 3. The table shows that the proposed PCKF-3 attains the lowest average RMSEs, whereas the EKF attains the highest. The others filters' average RMSEs lie between the Proposed PCKF-3 and the EKF.

Further, we compare the performance of the filters in terms of track divergence. The percentage of track loss of all the filters for $e_t = 50$ m, calculated over 10,000 MC runs are recorded in Table 4. From the table, we can observe that PCKF-3, PCKF-2,3t and PCKF-2 achieve the lowest percentage track loss whereas the EKF attains the highest. PCKF-3t, ICKF-3 and the CKF provide comparable track loss performances which is better than the PCKF-2t and the UKF. Lastly, we compare the filter performance in terms of their execution times. The execution time of all the filters relative to the EKF (considered to be unity) are provided in Table 4. It can be seen that the PCKF-3 is computationally more

TABLE 4 Track loss and relative execution time of different filters (Problem 1)

Filter	Track loss (%)	Relative exe. time
EKF	40.62	1.00
CKF	26.05	1.96
UKF	27.31	2.04
ICKF-3	25.52	2.04
PCKF-2t	27.22	1.92
PCKF-2	24.03	2.93
PCKF-3t	25.39	2.30
PCKF-2,3t	23.95	3.25
PCKF-3	23.90	9.35

Abbreviations: CKF, cubature Kalman filter; EKF, extended Kalman filter; ICKF, interpolatory CKF; PCKF, polynomial chaos Kalman filter; UKF, unscented Kalman filter.

expensive than the other filters. On the other hand, the PCKF-2t attains a comparable execution time as the CKF and the UKF. Also, the execution time of PCKF-3t is only slightly higher than the PCKF-2t.

Simulation result exhibits that the PCKF-2, PCKF-2,3t and PCKF-3 provide better track performance than all the other implemented filters. PCKF-2 and PCKF-2,3t can be a good choice for manoeuvring aircraft tracking problem due to high accuracy (compared to the EKF, the UKF and the CKF) and low computational burden (compared to PCKF-3).

Problem 2 In this problem, we consider a two sensors bearings-only tracking (BOT) problem [39], where the target follows a nearly straight line path with a nearly constant velocity. The measurement available from the sensors are bearing angles. The sensors are assumed to be stationary with coordinates (x_1, y_1) and (x_2, y_2) , respectively. The process equation is expressed as

$$\mathcal{X}_k = F\mathcal{X}_{k-1} + \eta_{k-1},$$

where the target state, $\mathcal{X} = [x_k \ v_{x_k} \ y_k \ v_{y_k}]^T$, (x_k, y_k) and (v_{x_k}, v_{y_k}) are the position and velocity of the target. The state transition matrix is $F = \begin{bmatrix} F_1 & 0_{2 \times 2} \\ 0_{2 \times 2} & F_1 \end{bmatrix}$, where $F_1 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ and $0_{2 \times 2}$ is a zero matrix of dimension 2.

The measurement equation becomes

$$\mathcal{Y}_k = \begin{bmatrix} \tan^{-1}((x_k - x_1)/(y_k - y_1)) \\ \tan^{-1}((x_k - x_2)/(y_k - y_2)) \end{bmatrix}.$$

The sensors are located at $(x_1, y_1) = (7700 \text{ m}, 9000 \text{ m})$ and $(x_2, y_2) = (6700 \text{ m}, 6000 \text{ m})$. The process noise is taken as $\eta_k \sim \mathcal{N}(0_{4 \times 1}, Q_k)$ with covariance $Q_k = \tilde{q} \begin{bmatrix} M & 0_{2 \times 2} \\ 0_{2 \times 2} & M \end{bmatrix}$,

where $M = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix}$. The process noise intensity is taken as $\tilde{q} = 9 \times 10^{-6} \text{ m}^2/\text{s}^3$. The measurement noise is taken as $\nu_k \sim \mathcal{N}(0_{2 \times 1}, R_k)$ with covariance $R_k = \text{diag}(\sigma_\theta^2, \sigma_\theta^2)$, where $\sigma_\theta = 3^\circ$. The sampling time (T) is assumed to be 1 s, and the simulation is lasted for 9 min. The initial truth of the target is $\mathcal{X}_0 = [9000 \text{ m} \ -5.144 \text{ m/s} \ 9000 \text{ m} \ -5.144 \text{ m/s}]^T$. The filters are initialised with the initial posterior state estimate $\hat{\mathcal{X}}_{0|0} = [10000 \text{ m} \ -7 \text{ m/s} \ 8000 \text{ m} \ -7 \text{ m/s}]^T$, and the initial error covariance. $P_{0|0} = \text{diag}([50000 \text{ m}^2 \ 300 \text{ m}^2/\text{s}^2 \ 30000 \text{ m}^2 \ 100 \text{ m}^2/\text{s}^2])$.

To solve this problem, we implement the EKF, CKF, ICKF-3, UKF and the proposed PCKFs (PCKF-2t, PCKF-2, PCKF-3t, PCKF-2,3t and PCKF-3). The tuning parameter of the UKF is set $\kappa = -1$ by following 3 - n [12]. As the process equation is linear, we use standard Kalman filter for the time update step and the measurement updated step is performed using the non-linear Gaussian filters. Figure 4 shows the stationary sensors location, the truth target trajectory and the estimated trajectory obtained by the proposed PCKF-3t for a single representative run. The figure shows that the proposed PCKF-3t successfully tracks the target. The range and velocity RMSE of the different filters excluding the track divergence ($e_t = 100 \text{ m}$) obtained from 500 MC runs are plotted in Figure 5a,b, respectively. The range RMSE plot shows that the proposed third-order chaos expansion-based filters (PCKF-3t and PCKF-3) attain the lowest RMSE, whereas the EKF has the highest RMSE. Other filters, such as the PCKF-2,3t, PCKF-2 and PCKF-2t, the UKF, the ICKF-3 and the CKF have comparable RMSE performances. The velocity RMSE plot shows that the PCKF-3 and PCKF-2,3t have the lowest RMSE, whereas the EKF attains the highest RMSE. Other filters provide comparable RMSEs and they lie between the EKF and PCKF-3. The slight differences in RMSEs among the various filters (except the EKF) are due to the plotting of RMSEs excluding the track loss criteria. The average RMSEs obtained from 500 MC runs of all the filters are listed in Table 5. From the table, we observe that the proposed PCKF-3 and PCKF-2,3t achieve the lowest average RMSE for both the position and the velocity.

Further, we compared the filters in terms of track loss and execution time, which are listed in Table 6. We calculate the percentage of track loss with track loss bound ($e_t = 100 \text{ m}$), from 10,000 MC runs. From the table, we see that the EKF has the highest percentage of track loss, whereas the PCKF-3t, PCKF-2,3t and PCKF-3 attain the lowest. Other filters such as the PCKF-2t, PCKF-2, CKF, ICKF-3 and UKF have similar track loss performance. The relative execution time of the different filters with respect to the EKF is provided in Table 6. From the table, we see that the PCKF-3, PCKF-2,3t and PCKF-2 have higher computation demand than the EKF. The proposed PCKF-3t requires a bit more execution time than the UKF.

The simulation result shows that the proposed PCKF-3t provides a much better estimation than the existing filters

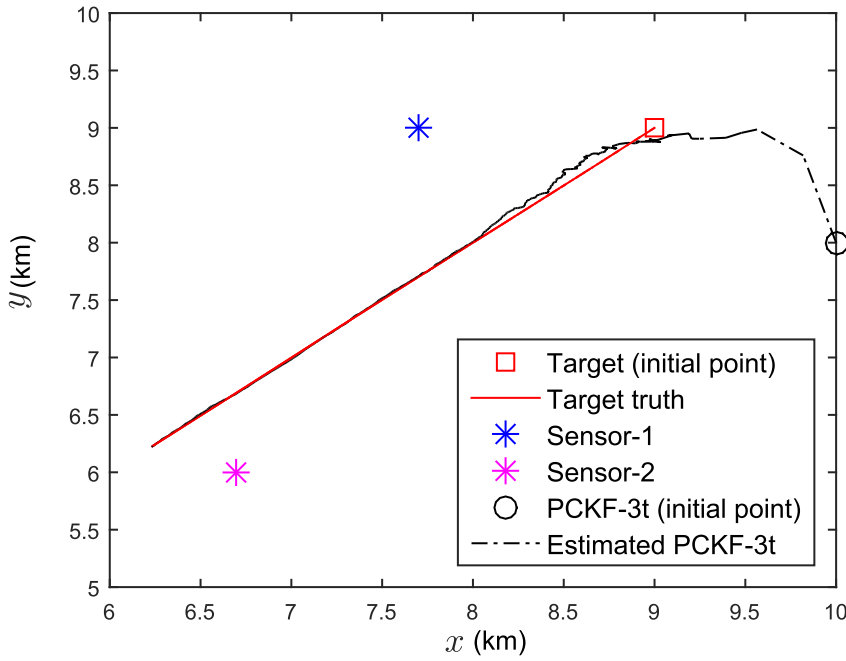


FIGURE 4 Truth and estimated trajectory for a single representative run. PCKF, polynomial chaos Kalman filter.

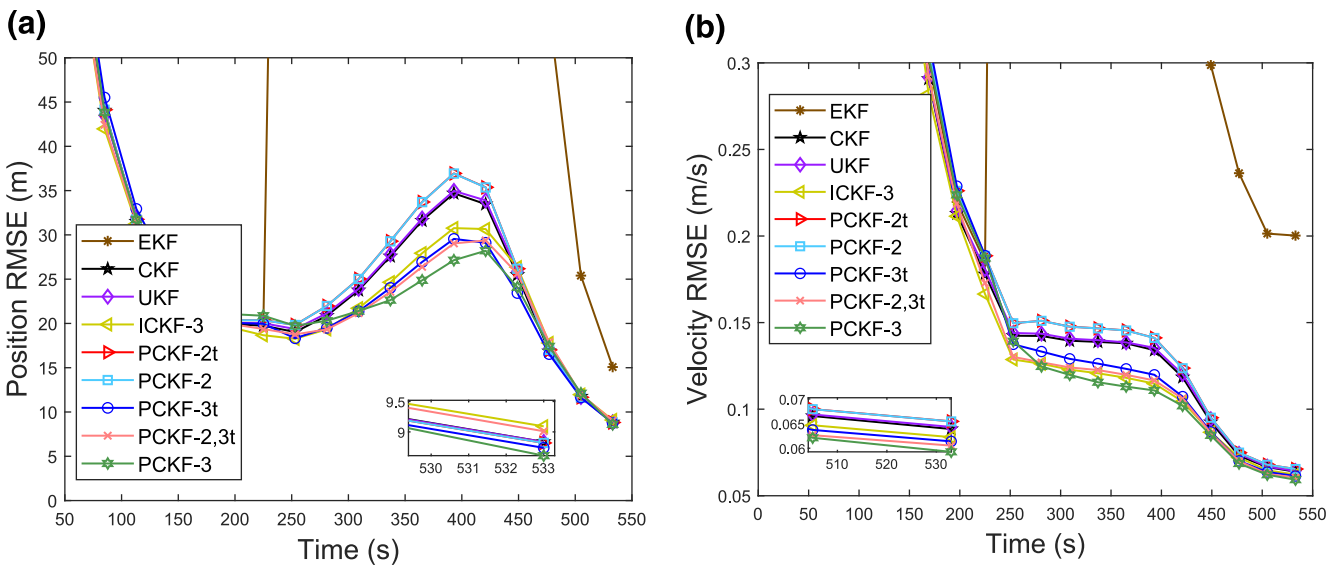


FIGURE 5 (a) Range RMSE; (b) Velocity RMSE of different filters obtained from 500 MC runs (problem 2). CKF, cubature Kalman filter; EKF, extended Kalman filter; ICKF, interpolatory CKF; PCKF, polynomial chaos Kalman filter; RMSE, root mean square error; UKF, unscented Kalman filter.

such as the EKF, CKF and UKF at a reasonable computation burden. Hence it can be a good choice for the practitioners for underwater BOT problem.

Based on the simulation results of both the target tracking experiments, we draw following conclusions:

- 1) The proposed PCKF-2t provides filtering performance comparable to the CKF and the UKF for both the problems. The computational cost of the PCKF-2t is also comparable to the CKF and UKF.
- 2) The PCKF-3 outperforms all the other filters for both the target tracking problems. But its computational time is higher than other filters.
- 3) The PCKF-3t, PCKF-2 and PCKF-2,3t provide similar or better estimation results than the other quadrature filters at a reasonable added computation cost. Hence either of them can be a good choice for on-board applications.
- 4) From the simulation results of both the target tracking problems, we see that among the proposed filters, PCKF-3 achieves the best estimation accuracy whereas the PCKF-2t

TABLE 5 Average RMSE in position and velocity of different filters (problem 2)

Filter	Position (m)	Velocity (m/s)
EKF	93.95	1.95
CKF	41.21	1.69
UKF	41.36	1.69
ICKF-3	39.75	1.67
PCKF-2t	41.42	1.61
PCKF-2	41.10	1.60
PCKF-3t	39.23	1.48
PCKF-2,3t	39.15	1.47
PCKF-3	39.00	1.45

Abbreviations: CKF, cubature Kalman filter; EKF, extended Kalman filter; ICKF, interpolatory CKF; PCKF, polynomial chaos Kalman filter; RMSE, root mean square error; UKF, unscented Kalman filter.

TABLE 6 Percentage of track loss and relative execution time of different filters (problem 2)

Filter	Track loss (%)	Relative exe. time
EKF	28.57	1.00
CKF	5.10	2.11
UKF	6.59	2.21
ICKF-3	4.42	2.21
PCKF-2t	5.57	2.43
PCKF-2	5.56	3.17
PCKF-3t	0.13	2.95
PCKF-2,3t	0.12	3.55
PCKF-3	0.12	5.81

Abbreviations: CKF, cubature Kalman filter; EKF, extended Kalman filter; ICKF, interpolatory CKF; PCKF, polynomial chaos Kalman filter; UKF, unscented Kalman filter.

attains the least. On the other hand, the computation demand of PCKF-2t is the least whereas PCKF-3 has the highest. If computational budget allow us, then the PCKF-3 can be a good choice. PCKF-3t, PCKF-2 and PCKF-2,3t may offer a good compromise between the computation cost and estimation accuracy as well as track loss performance.

6 | DISCUSSION AND CONCLUSION

This paper proposed a Gaussian state estimation algorithm based on polynomial chaos approximation of non-linear dynamics for the target tracking applications. The process and measurement update are performed with the help of the polynomial chaos expansion and a set of CPs. We used up to third-order polynomial chaos expansion to derive the filtering algorithm. For a high-dimensional system, the proposed

filters (PCKF-2 and PCKF-3) use a large number of CPs, which eventually increases their computation cost. The computation complexity of the proposed PCKF is reduced by removing the cross terms from the chaos expansion. As a result, a few variants of it, namely PCKF-2t, PCKF-3t and PCKF-2,3t, are developed. The performances of the proposed filters (PCKF-2t, PCKF-2, PCKF-3t, PCKF-2,3t and PCKF-3) are compared with the EKF, the CKF, the ICKF-3 and the UKF for two target tracking problems. The proposed PCKFs provide better performance than the EKF and better or comparable performance with other Gaussian filters for both the simulated problems at small additional increment of the computation cost. The high execution time of the proposed filter is a drawback, which may prove to be a deterrent in using the proposed filter for a high dimensional system such as GNSS positioning and tracking. PCKF-2,3t and PCKF-3 can be good choice if the computational budget allows it; PCKF-3t may offer a good compromise between computational speed and estimation accuracy as well as track loss performance. The proposed filters thus increase the choice available to engineers when it comes to designing on-board filters in order to achieve specific computational complexity versus filtering performance trade-off. Exploration of the proposed PCKFs under the model and noise mismatch scenario remains a topic for potential future research.

AUTHOR CONTRIBUTION

Kundan Kumar: Conceptualisation; formal analysis; methodology; software; validation; visualisation; writing – original draft. **Ranjeet Kumar Tiwari:** Conceptualisation; formal analysis; methodology; validation; visualisation; writing – review and editing. **Shovan Bhaumik:** Conceptualisation; formal analysis; methodology; project administration; supervision; validation; visualisation; writing – review and editing. **Paresh Date:** Conceptualisation; formal analysis; funding acquisition; methodology; supervision; validation; visualisation; writing – review and editing.

ACKNOWLEDGEMENT

The authors are grateful to the anonymous reviewers for their useful suggestions, which help to uplift the quality of the paper substantially.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

This research did not use any experimentally generated data or data from any publicly available dataset. Model definitions (including specified probability distributions) and parameter values (including the initialisation parameters) provided in the paper are adequate for reproducing the qualitative behaviour of algorithms illustrated in the paper.

ORCID

Kundan Kumar  <https://orcid.org/0000-0002-2977-8815>

REFERENCES

1. Ristic, B., Arulampalam, S., Gordon, N.: *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, London (2003)
2. Leong, P.H., et al.: A Gaussian-sum based cubature Kalman filter for bearings-only tracking. *IEEE Trans. Aero. Electron. Syst.* 49(2), 1161–1176 (2013). <https://doi.org/10.1109/taes.2013.6494405>
3. Li, X.R., Jilkov, V.P.: Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Trans. Aero. Electron. Syst.* 41(4), 1255–1321 (2005). <https://doi.org/10.1109/taes.2005.1561886>
4. Lin, X., et al.: Comparison of EKF, pseudomeasurement, and particle filters for a bearing-only target tracking problem. In: *Signal and Data Processing of Small Targets 2002*, vol. 4728, pp. 240–250. SPIE (2002)
5. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, New York (2004)
6. Särkkä, S.: *Bayesian Filtering and Smoothing*. Cambridge University Press, Cambridge (2013)
7. Miller, R.N., Ghil, M., Gauthiez, F.: Advanced data assimilation in strongly nonlinear dynamical systems. *J. Atmos. Sci.* 51(8), 1037–1056 (1994). [https://doi.org/10.1175/1520-0469\(1994\)051<1037:adaisn>2.0.co;2](https://doi.org/10.1175/1520-0469(1994)051<1037:adaisn>2.0.co;2)
8. Bhaumik, S., Date, P.: *Nonlinear Estimation: Methods and Applications with Deterministic Sample Points*. CRC Press, Boca Raton (2019)
9. Arasaratnam, I., Haykin, S.: Cubature Kalman filters. *IEEE Trans. Automat. Control.* 54(6), 1254–1269 (2009). <https://doi.org/10.1109/tac.2009.2019800>
10. Zhang, Y., et al.: Interpolatory cubature Kalman filters. *IET Control Theor. Appl.* 9(11), 1731–1739 (2015). <https://doi.org/10.1049/iet-cta.2014.0873>
11. Zhang, Y., et al.: Embedded cubature Kalman filter with adaptive setting of free parameter. *Signal Process.* 114, 112–116 (2015). <https://doi.org/10.1016/j.sigpro.2015.02.022>
12. Julier, S.J., Uhlmann, J.K.: New extension of the Kalman filter to nonlinear systems. In: *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, pp. 182–193. International Society for Optics and Photonics (1997)
13. Julier, S., Uhlmann, J., Durrant-Whyte, H.F.: A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Automat. Control.* 45(3), 477–482 (2000). <https://doi.org/10.1109/9.847726>
14. Ito, K., Xiong, K.: Gaussian filters for nonlinear filtering problems. *IEEE Trans. Automat. Control.* 45(5), 910–927 (2000). <https://doi.org/10.1109/9.855552>
15. Arulampalam, M.S., et al.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* 50(2), 174–188 (2002). <https://doi.org/10.1109/78.978374>
16. Wiener, N.: The homogeneous chaos. *Am. J. Math.* 60(4), 897–936 (1938). <https://doi.org/10.2307/2371268>
17. Xiu, D., Karniadakis, G.E.: The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* 24(2), 619–644 (2002). <https://doi.org/10.1137/s1064827501387826>
18. Ren, Z., et al.: Probabilistic power flow analysis based on the stochastic response surface method. *IEEE Trans. Power Syst.* 31(3), 2307–2315 (2015). <https://doi.org/10.1109/tpwrs.2015.2461159>
19. Ghanem, R.G., Spanos, P.D.: *Stochastic Finite Elements: A Spectral Approach*. Dover Publications, New York (2003)
20. Yu, Z., Cui, P., Ni, M.: A polynomial chaos based square-root Kalman filter for mars entry navigation. *Aero. Sci. Technol.* 51, 192–202 (2016). <https://doi.org/10.1016/j.ast.2016.02.009>
21. Dutta, P., Bhattacharya, R.: Nonlinear estimation with polynomial chaos and higher order moment updates. In: *Proceedings of the 2010 American Control Conference*, pp. 3142–3147. IEEE (2010)
22. Li, J., Xiu, D.: A generalized polynomial chaos based ensemble Kalman filter with high accuracy. *J. Comput. Phys.* 228(15), 5454–5469 (2009). <https://doi.org/10.1016/j.jcp.2009.04.029>
23. Xu, Y., Mili, L., Zhao, J.: A novel polynomial-chaos-based Kalman filter. *IEEE Signal Process. Lett.* 26(1), 9–13 (2018). <https://doi.org/10.1109/lsp.2018.2879453>
24. Li, H., Zhang, D.: Probabilistic collocation method for flow in porous media: comparisons with other stochastic methods. *Water Resour. Res.* 43(9) (2007). <https://doi.org/10.1029/2006wr005673>
25. Li, D.Q., et al.: A comparative study of three collocation point methods for odd order stochastic response surface method. *Struct. Eng. Mech.* 45(5), 595–611 (2013). <https://doi.org/10.12989/sem.2013.45.5.595>
26. Bhaumik, S., Swati: Cubature quadrature Kalman filter. *IET Signal Process.* 7(7), 533–541 (2013). <https://doi.org/10.1049/iet-spr.2012.0085>
27. Isukapalli, S.S.: *Uncertainty Analysis of Transport Transformation Models*. PhD Thesis, Rutgers University, New Brunswick (1999)
28. Haesen, E., et al.: A probabilistic formulation of load margins in power systems with stochastic generation. *IEEE Trans. Power Syst.* 24(2), 951–958 (2009). <https://doi.org/10.1109/tpwrs.2009.2016525>
29. Haug, A.J.: *Bayesian Estimation and Tracking: A Practical Guide*. John Wiley & Sons, Hoboken (2012)
30. Eldred, M.: Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design. In: *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA*, pp. 2274 (2009)
31. Ng, L.W.-T., Eldred, M.: Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation. In: *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, pp. 1852 (2012)
32. Hosder, S., Walters, R., Balch, M.: Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. In: *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pp. 1939 (2007)
33. Xu, J., Kong, F.: A cubature collocation based sparse polynomial chaos expansion for efficient structural reliability analysis. *Struct. Saf.* 74, 24–31 (2018). <https://doi.org/10.1016/j.strusafe.2018.04.001>
34. Wang, C., Matthies, H.G.: Evidence theory-based reliability optimization design using polynomial chaos expansion. *Comput. Methods Appl. Mech. Eng.* 341, 640–657 (2018). <https://doi.org/10.1016/j.cma.2018.07.015>
35. Karlsson, R., Schon, T., Gustafsson, F.: Complexity analysis of the marginalized particle filter. *IEEE Trans. Signal Process.* 53(11), 4408–4411 (2005). <https://doi.org/10.1109/tsp.2005.857061>
36. Arasaratnam, I., Haykin, S.: Square-root quadrature Kalman filtering. *IEEE Trans. Signal Process.* 56(6), 2589–2593 (2008). <https://doi.org/10.1109/tsp.2007.914964>
37. Boers, Y., Driessen, J.N.: Interacting multiple model particle filter. *IEEE Proc. Radar, Sonar Navig.* 150(5), 344–349 (2003). <https://doi.org/10.1049/ip-rsn:20030741>
38. Julier, S.J.: The scaled unscented transformation. In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 6, pp. 4555–4559. IEEE (2002)
39. Chang, D.C., Fan, M.W.: Interacting multiple model particle filtering using new particle resampling algorithm. In: *2014 IEEE Global Communications Conference*, pp. 3215–3219. IEEE (2014)

How to cite this article: Kumar, K., et al.: Polynomial chaos Kalman filter for target tracking applications. *IET Radar Sonar Navig.* 1–14 (2022). <https://doi.org/10.1049/rsn2.12338>