

Polynomial Chaos Expansion Based Rauch–Tung–Striebel Smoothers

Kundan Kumar

Department of Electrical Engineering
and Automation
Aalto University, Finland
Email: kundan.kumar@aalto.fi

Simo Särkkä

Department of Electrical Engineering
and Automation
Aalto University, Finland
Email: simo.sarkka@aalto.fi

Abstract—This article introduces Gaussian approximation-based smoothing algorithms for nonlinear stochastic state space models using the polynomial chaos expansion (PCE). Initially, we present a smoothing algorithm, where the nonlinear functions of the state space model are approximated using a PCE that is formed using a set of collocation points generated from the filtering distribution. Subsequently, an iterative variant of the proposed smoothing algorithm is also presented. It iteratively forms a PCE approximation to the nonlinear functions by using collocation points generated from the current posterior approximation. The performance of the algorithms is evaluated on pendulum and aircraft tracking problems.

Index Terms—Gaussian approximation-based smoother, polynomial chaos expansion, point collocation, iterative smoother.

I. INTRODUCTION

In this paper, we introduce an approximate polynomial chaos expansion (PCE) based smoothing algorithm for nonlinear state space models. The smoothing solutions for the nonlinear state space models find applications in various real-life scenarios such as target tracking, navigation, guidance systems, audio, and biomedical signal processing [1]–[5]. We consider dynamic systems that can be expressed as stochastic state space models of the following form [1], [2]:

$$x_k = f(x_{k-1}) + \eta_{k-1}, \quad (1)$$

$$y_k = h(x_k) + \nu_k, \quad (2)$$

where $x_k \in \mathbb{R}^{n_x}$, and $y_k \in \mathbb{R}^{n_y}$ are the state of the system and the sensor measurement, respectively. Above, $f(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ and $h(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ are known nonlinear functions. The process noise, η_{k-1} and the measurement noise, ν_k are assumed uncorrelated white Gaussian with mean zero, and covariances Q_{k-1} and R_k , respectively. The initial state $x_0 \sim \mathcal{N}(\hat{x}_{0|0}, P_{0|0})$, η_{k-1} , and ν_k are mutually independent of each other.

Smoothing involves estimating the past state history of a dynamic system by utilizing all the available measurements, leading to enhanced estimation performance over filtering. The Bayesian smoother computes the marginal posterior distribution of the state, x_k given measurements up to time step T , $y_{1:T}$, that is, $p(x_k | y_{1:T})$, $k \in \{1, \dots, T\}$ in two steps: (i) forward pass, and (ii) backward pass. The forward pass

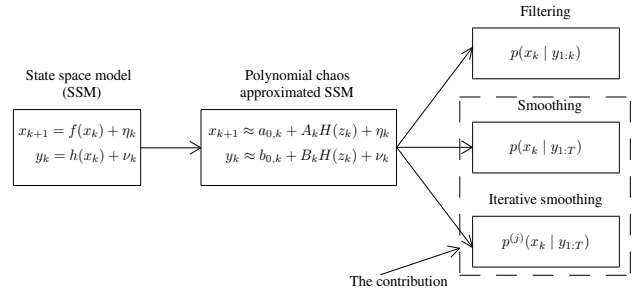


Fig. 1 Illustration of the basic idea behind the proposed chaos expansion-based smoother and its iterative variant. Initially, we approximate the stochastic state space model using the polynomial chaos expansion through a weighted sum of orthogonal polynomial basis functions. Subsequently, a smoothing algorithm and its iterative variants are developed based on the approximated state space model.

consists of filtering to recursively compute $p(x_k | y_{1:k})$ [1], [2]

$$p(x_k | y_{1:k}) \propto p(y_k | x_k) p(x_k | y_{1:k-1}),$$

$$p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1}.$$

In the backward pass, $p(x_k | y_{1:T})$ is being computed recursively backwards, starting from $k = T$ via

$$p(x_k | y_{1:T}) = p(x_k | y_{1:k}) \times \int \left[\frac{p(x_{k+1} | x_k) p(x_{k+1} | y_{1:T})}{p(x_{k+1} | y_{1:k})} \right] dx_{k+1}.$$

For linear Gaussian systems, the distributions remain Gaussian, and a closed-form solution of them can be computed exactly using the Rauch–Tung–Striebel (RTS) smoother [2], [6], [7]. However, for nonlinear systems, the distributions lose their Gaussian property, and closed-form solutions are not available. To address this, various methods exist to obtain approximate solutions. Often in literature, these distributions are approximated as Gaussian [2], [8]–[11] by using moment matching, which is also the approach that we use here. Other methods also exist in the literature, including the sequential Monte Carlo and related methods [12].

Several Gaussian approximation-based smoothers for nonlinear systems exist in the literature (see, e.g., [2]). The clas-

sical one is the extended RTS smoother (ERTSS), which uses the local linearization based on the Taylor series expansion [2], [7], [13]. Various other Gaussian approximation-based smoothers have been developed, including the cubature RTS [14], unscented RTS [11], Gauss–Hermite RTS [15], and the Fourier–Hermite RTS [16], among others. To further enhance estimation accuracy, iterative variants have been developed [2], [17].

Numerous estimation algorithms have also been developed based on the polynomial chaos expansion (PCE) [18], [19]. Examples include the polynomial chaos extended Kalman filter (PCEKF) [20], polynomial chaos ensemble KF (PCEnKF) [21], and polynomial chaos KF [22], [23]. In this paper, we extend the methods developed in [23] to derive novel smoothing algorithms. The developed algorithms utilize the PCE to approximate the state space model through a weighted sum of orthogonal polynomial basis functions, a concept initially introduced in [18]. Subsequently, the integrals associated with the algorithm are approximately evaluated by fitting the polynomial to the nonlinear functions by using a set of collocation points (CPs). A bit similar approach has been adopted in the Fourier–Hermite series-based algorithms [16], [24], but based on spectral expansions.

The main contributions of this article are (1) to introduce a Gaussian approximation-based smoothing algorithm using a polynomial chaos expansion and (2) an iterative variant of it utilizing an iterated posterior distribution approximation. We also numerically illustrate the performance of the proposed methods. The proposed approach is illustrated in Fig. 1.

II. POLYNOMIAL CHAOS EXPANSION

In this paper, the polynomial chaos expansion is employed to approximate the nonlinear transition functions, $f(x)$, where $x \sim \mathcal{N}(\hat{x}, P)$. Utilizing the affine transformation [23] $x = \hat{x} + Sz$, we transform the random variable x in to a standard Gaussian random variable z , that is, $z \sim \mathcal{N}(0, I)$, where $P = SS^\top$. The d -th order chaos expansion of the function can be expressed as [19], [23]

$$\begin{aligned} f(\hat{x} + Sz) &\approx a_0 + \sum_{i_1=1}^{n_x} a_{i_1} H_1(z_{i_1}) + \sum_{i_1=1}^{n_x} \sum_{i_2=1}^{i_1} a_{i_1 i_2} \\ &H_2(z_{i_1}, z_{i_2}) + \sum_{i_1=1}^{n_x} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} H_3(z_{i_1}, z_{i_2}, z_{i_3}) + \cdots \\ &+ \sum_{i_1=1}^{n_x} \sum_{i_2=1}^{i_1} \cdots \sum_{i_d=1}^{i_{d-1}} a_{i_1 i_2 \cdots i_d} H_d(z_{i_1}, z_{i_2}, \dots, z_{i_d}), \end{aligned} \quad (3)$$

where $a_{i_1 \dots i_s}$ are the unknown PCE coefficients, the random variable $z = [z_1 \ z_2 \ \cdots \ z_{n_x}]^\top$, $m = \binom{n_x+d}{d}$ is the total number of coefficients, and $H_d(\cdot)$ is the d -th degree multidimensional Hermite polynomial [19],

$$\begin{aligned} H_d(z_1, z_2, \dots, z_{n_x}) \\ = (-1)^d \exp\left(\frac{1}{2} z^\top z\right) \frac{\partial^{n_x}}{\partial z_1 \partial z_2 \cdots \partial z_{n_x}} \exp\left(-\frac{1}{2} z^\top z\right). \end{aligned}$$

The Hermite polynomials are orthogonal under the inner product

$$\langle f, g \rangle = \int f(z) g(z) \mathcal{N}(z | 0, I) dz = E[fg],$$

that is $E[H_p H_q] = 0$ if $p \neq q$, and here, the expectation is being computed with respect to the standard Gaussian distribution. After rearranging and simplifying, we can express Eq. (3) in a more concise way as

$$f(\hat{x} + Sz) \approx a_0 + AH(z), \quad (4)$$

where $a_0 \in \mathbb{R}^{n_x}$, and the matrices A and $H(z)$ are

$$\begin{aligned} A_{n_x \times (m-1)} &= [a_1 \ a_2 \ \cdots \ a_{n_x \cdots n_x}], \\ H(z)_{(m-1) \times 1} &= [H_1(z_1) \ \cdots \ H_d(z_{n_x}, \dots, z_{n_x})]^\top. \end{aligned}$$

Utilizing the aforementioned approximation, the stochastic state space models in Eqs. (1)-(2) can be expressed as

$$x_k \approx a_{0,k-1} + A_{k-1} H(z_{k-1}) + \eta_{k-1}, \quad (5)$$

$$y_k \approx b_{0,k} + B_k H(z_k) + \nu_k, \quad (6)$$

where $b_0 \in \mathbb{R}^{n_y}$ and $B \in \mathbb{R}^{n_y \times m-1}$ are the coefficient matrices, a_0, A and $H(z)$ are as defined above.

Remark 1: For a standard Gaussian random variable z , the matrix $H(z)$ exhibits the following properties: $E[H(z)] = 0$ and $E[H(z)H(z)^\top] = I$.

A. Determining the unknown coefficient

Here, we discuss the evaluation of the unknown coefficient matrices of the approximation. Eq. (4) can be rewritten as

$$f(\hat{x} + Sz) = [a_0 \ A] \begin{bmatrix} 1 \\ H(z) \end{bmatrix} = A' H'(z). \quad (7)$$

To determine the matrix A' , consisting of the coefficients a_0 and A , we utilize a method based on collocation points (CPs) $\xi_i \in \mathbb{R}^{n_x}$ for $i = 1, \dots, m$ [22], [23]. Evaluating both sides of Eq. (7) on these CPs, the equation becomes

$$\chi = \mathbf{H} A'^\top, \quad (8)$$

where the matrices $\chi \in \mathbb{R}^{m \times n_x}$ and $\mathbf{H} \in \mathbb{R}^{m \times m}$ are

$$\chi = \begin{bmatrix} f^\top(\hat{x} + S\xi_1) \\ f^\top(\hat{x} + S\xi_2) \\ \vdots \\ f^\top(\hat{x} + S\xi_m) \end{bmatrix}, \mathbf{H} = \begin{bmatrix} H'^\top(\xi_1) \\ H'^\top(\xi_2) \\ \vdots \\ H'^\top(\xi_m) \end{bmatrix}. \quad (9)$$

Solving the linear equation in Eq. (8), we get the coefficient matrix

$$A'^\top = \mathbf{H}^{-1} \chi. \quad (10)$$

If the matrix, \mathbf{H} is not invertible, we can use a (regularized) pseudo-inverse to solve the equation. Similarly, we can evaluate the coefficient matrix, $B' = [b_0 \ B]$ for the measurement model in Eq. (6). Here, the collocation points are selected as in [22], [23]. A pseudo-code for PCE is provided in Algorithm 1.

Algorithm 1 PCE using collocation points

```

1: function  $A' = \text{PCE}(f(\cdot), \hat{x}, P, \xi, \mathbf{H})$ .
2:   Compute the square root  $S$  of  $P = SS^\top$ .
3:   for  $i = 1, \dots, m$  do
4:     Form the translated and scaled collocation point
        $\psi_i = \hat{x} + S\xi_i$ .
5:   Evaluate the function at the point  $\chi_i = f(\psi_i)$ .
6:   end for
7:   Form the matrix,  $\chi = [\chi_1^\top \ \chi_2^\top \ \dots \ \chi_m^\top]^\top$ .
8:   Compute the coefficient matrix  $A' = (\mathbf{H}^{-1}\chi)^\top$ .
9: end function

```

III. POLYNOMIAL CHAOS RTS SMOOTHER

In this section, we develop the RTS smoother for the state space model in Eqs. (5)-(6) utilizing the polynomial chaos expansion. As discussed in Sec. I, it is performed in two steps: (i) forward pass and (ii) backward pass.

A. Forward pass

The forward pass consists of the filtering algorithm. Assuming that both $p(x_k | x_{k-1})$ and $p(x_{k-1} | y_{1:k-1})$ are Gaussian distributions, the joint distribution of x_{k-1} and x_k given $y_{1:k-1}$ can be expressed as

$$\begin{aligned}
p(x_{k-1}, x_k | y_{1:k-1}) &= p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) \\
&\approx \mathcal{N}(x_k | a_{0,k-1} + A_{k-1}H(z_{k-1}), Q_{k-1}) \\
&\quad \mathcal{N}(x_{k-1} | \hat{x}_{k-1|k-1}, P_{k-1|k-1}) \\
&\approx \mathcal{N}\left(\begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix} \middle| \hat{\mathcal{X}}_{k-1}, \mathcal{P}_{k-1}\right),
\end{aligned} \tag{11}$$

where moment matching gives

$$\begin{aligned}
\hat{\mathcal{X}}_{k-1} &= \begin{bmatrix} \hat{x}_{k-1|k-1} \\ a_{0,k-1} \end{bmatrix}, \\
\mathcal{P}_{k-1} &= \begin{bmatrix} P_{k-1|k-1} & S_{k-1|k-1} \bar{I} A_{k-1}^\top \\ A_{k-1} \bar{I}^\top S_{k-1|k-1}^\top & A_{k-1} A_{k-1}^\top + Q_{k-1} \end{bmatrix},
\end{aligned} \tag{12}$$

and $\bar{I} = [I \ 0]$. It is worth mentioning again here that the random variable z_{k-1} follows a standard Gaussian distribution.

Following [2, lemma A.3], the marginal distribution of x_k is

$$p(x_k | y_{1:k-1}) \approx \mathcal{N}(x_k | \hat{x}_{k|k-1}, P_{k|k-1}),$$

where

$$\begin{aligned}
\hat{x}_{k|k-1} &= a_{0,k-1}, \\
P_{k|k-1} &= A_{k-1} A_{k-1}^\top + Q_{k-1}.
\end{aligned}$$

We can then approximate the joint distribution of x_k and y_k given $y_{1:k-1}$ as Gaussian as follows:

$$\begin{aligned}
p(x_k, y_k | y_{1:k-1}) &= p(y_k | x_k) p(x_k | y_{1:k-1}) \\
&\approx \mathcal{N}(y_k | b_{0,k} + B_k H(z_k), R_k) \mathcal{N}(x_k | \hat{x}_{k|k-1}, P_{k|k-1}) \\
&\approx \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix} \middle| \begin{bmatrix} \hat{x}_{k|k-1} \\ b_{0,k} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & A_{k-1} B_k^\top \\ B_k A_{k-1}^\top & B_k B_k^\top + R_k \end{bmatrix}\right).
\end{aligned}$$

By [2, lemma A.3], the conditional distribution of x_k is

$$p(x_k | y_k, y_{1:k-1}) = p(x_k | y_{1:k}) \approx \mathcal{N}(x_k | \hat{x}_{k|k}, P_{k|k}),$$

where

$$\begin{aligned}
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + A_{k-1} B_k^\top (B_k B_k^\top + R_k)^{-1} (y_k - b_{0,k}), \\
P_{k|k} &= P_{k|k-1} - A_{k-1} B_k^\top (B_k B_k^\top + R_k)^{-1} B_k A_{k-1}^\top.
\end{aligned}$$

B. Backward pass

After computing $p(x_k | y_{1:k})$, $k \in \{1, \dots, T\}$, the backward pass is performed to evaluate $p(x_k | y_{1:T})$ recursively backwards starting from $k = T$. Similarly to Eq. (11), the joint distribution of the states x_k and x_{k+1} given $y_{1:k}$ can be approximated as

$$\begin{aligned}
p(x_k, x_{k+1} | y_{1:k}) &= p(x_{k+1} | x_k) p(x_k | y_{1:k}) \\
&\approx \mathcal{N}(x_{k+1} | a_{0,k} + A_k H(z_k), Q_k) \mathcal{N}(x_k | \hat{x}_{k|k}, P_{k|k}) \\
&\approx \mathcal{N}\left(\begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} \middle| \begin{bmatrix} \hat{x}_{k|k} \\ a_{0,k} \end{bmatrix}, \begin{bmatrix} P_{k|k} & S_{k|k} \bar{I} A_k^\top \\ A_k \bar{I}^\top S_{k|k}^\top & A_k A_k^\top + Q_k \end{bmatrix}\right).
\end{aligned}$$

Due to the Markov property of the states, we have

$$p(x_k | x_{k+1}, y_{1:T}) = p(x_k | x_{k+1}, y_{1:k}) \approx \mathcal{N}(x_k | \hat{\mathcal{X}}'_k, \mathcal{P}'_k),$$

where

$$K_s = S_{k|k} \bar{I} A_k^\top P_{k+1|k}^{-1}, \tag{13}$$

$$\hat{\mathcal{X}}'_k = \hat{x}_{k|k} + K_s (x_{k+1} - \hat{x}_{k+1|k}), \tag{14}$$

$$\mathcal{P}'_k = P_{k|k} - K_s P_{k+1|k} K_s^\top. \tag{15}$$

Assuming that the smoothing distribution $p(x_{k+1} | y_{1:T})$ is Gaussian distribution, the joint distribution of x_k and x_{k+1} conditioned on $y_{1:T}$ can be expressed as follows:

$$\begin{aligned}
p(x_{k+1}, x_k | y_{1:T}) &= p(x_k | x_{k+1}, y_{1:T}) p(x_{k+1} | y_{1:T}) \\
&\approx \mathcal{N}(x_k | \hat{\mathcal{X}}'_k, \mathcal{P}'_k) \mathcal{N}(x_{k+1} | \hat{x}_{k+1|T}^s, P_{k+1|T}^s) \\
&\approx \mathcal{N}\left(\begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix} \middle| \hat{\mathcal{X}}''_k, \mathcal{P}''_k\right),
\end{aligned}$$

where

$$\hat{\mathcal{X}}''_k = \begin{bmatrix} \hat{x}_{k+1|T}^s \\ \hat{x}_{k|k} + K_s (\hat{x}_{k+1|T}^s - \hat{x}_{k+1|k}) \end{bmatrix},$$

$$\mathcal{P}''_k = \begin{bmatrix} P_{k+1|T}^s & P_{k+1|T}^s K_s^\top \\ K_s P_{k+1|T}^s & P_{k|k} + K_s (P_{k+1|T}^s - P_{k+1|k}) K_s^\top \end{bmatrix}.$$

The marginal distribution of x_k can then be expressed as

$$p(x_k | y_{1:T}) \approx \mathcal{N}(x_k | \hat{x}_{k|T}^s, P_{k|T}^s),$$

where

$$\begin{aligned}
\hat{x}_{k|T}^s &= \hat{x}_{k|k} + K_s (\hat{x}_{k+1|T}^s - \hat{x}_{k+1|k}), \\
P_{k|T}^s &= P_{k|k} + K_s (P_{k+1|T}^s - P_{k+1|k}) K_s^\top.
\end{aligned}$$

The resulting PCRTS smoother is given in Algorithm 2.

Algorithm 2 PCRTS smoother

```

1: function  $[\hat{x}_{k|T}^s, P_{k|T}^s] = \text{PCRTS}(\hat{x}_{0|0}, P_{0|0}, \xi, \mathbf{H})$ .
2: for  $k = 1, \dots, T$  do
3:    $[a_{0,k-1} \ A_{k-1}] =$ 
      $\text{PCE}(f(\cdot), \hat{x}_{k-1|k-1}, P_{k-1|k-1}, \xi, \mathbf{H})$ .
4:    $\hat{x}_{k|k-1} = a_{0,k-1}$ .
5:    $P_{k|k-1} = A_{k-1} A_{k-1}^\top + Q_{k-1}$ .
6:    $[b_{0,k} \ B_k] = \text{PCE}(h(\cdot), \hat{x}_{k|k-1}, P_{k|k-1}, \xi, \mathbf{H})$ .
7:    $\hat{x}_{k|k} = \hat{x}_{k|k-1} + A_{k-1} B_k^\top$ 
      $\times (B_k B_k^\top + R_k)^{-1} (y_k - b_{0,k})$ .
8:    $P_{k|k} = P_{k|k-1} - A_{k-1} B_k^\top (B_k B_k^\top + R_k)^{-1} B_k A_{k-1}^\top$ .
9: end for
10:  $\hat{x}_{T|T}^s = \hat{x}_{T|T}$  and  $P_{T|T}^s = P_{T|T}$ .
11: for  $k = T-1, \dots, 1$  do
12:   Compute the square root  $S_{k|k}$  of  $P_{k|k}$ .
13:    $K_s = S_{k|k} [I \ 0] A_k^\top P_{k+1|k}^{-1}$ .
14:    $\hat{x}_{k+1|T}^s = \hat{x}_{k+1|k} + K_s (\hat{x}_{k+1|T}^s - \hat{x}_{k+1|k})$ .
15:    $P_{k+1|T}^s = P_{k+1|k} + K_s (P_{k+1|T}^s - P_{k+1|k}) K_s^\top$ .
16: end for
17: end function

```

C. Iterative PCRTS smoother

We can also develop an iterative extension of PCRTSS by using a similar approach as in the posterior linearization smoother [17], which we refer to as iterative PCRTSS (IPCRTSS). The basic idea behind IPCRTSS is to perform PCE with respect to the smoothing distribution,

$$p(x_k | y_{1:T}) \approx \mathcal{N}(x_k | \hat{x}_{k|T}^s, P_{k|T}^s).$$

This can be done by evaluating the translated and scaled collocation points as

$$\psi_i^{(j)} = \hat{x}^{s(j)} + S^{s(j)} \xi_i, \quad i = 1, \dots, m,$$

where $\hat{x}^{s(j)}$ and $S^{s(j)}$ are the mean and square root of the covariance of the current posterior smoothing distribution approximation $x \sim \mathcal{N}(\hat{x}^{s(j)}, P^{s(j)})$ at iteration j . Eq. (8) now becomes

$$\chi^{(j)} = \mathbf{H}^{(j)} A'^\top, \quad (16)$$

where

$$\mathbf{H}^{(j)} = \begin{bmatrix} H'^\top ([S^{(j)}]^{-1} (\hat{x}^{s(j)} + S^{s(j)} \xi_1 - \hat{x}^{(j)})) \\ H'^\top ([S^{(j)}]^{-1} (\hat{x}^{s(j)} + S^{s(j)} \xi_2 - \hat{x}^{(j)})) \\ \vdots \\ H'^\top ([S^{(j)}]^{-1} (\hat{x}^{s(j)} + S^{s(j)} \xi_m - \hat{x}^{(j)})) \end{bmatrix}, \quad (17)$$

and $\hat{x}^{(j)}$ and $S^{(j)}$ are the current filtering mean and square root of the covariance. The matrix B' can be computed analogously. The smoother now becomes an iterative method, which is provided in Algorithm 3.

IV. SIMULATION RESULTS

To evaluate the performance of the developed smoothers, we conducted experiments on (i) pendulum and (ii) aircraft

Algorithm 3 Iterative PCRTS smoother

```

1: function  $[\hat{x}_{k|T}^{s(N)}, P_{k|T}^{s(N)}] = \text{IPCRTS}(\hat{x}_{0|0}^{s(1)}, P_{0|0}^{s(1)}, \xi, \mathbf{H})$ .
2: Initialization: Compute  $\hat{x}_{k|T}^{s(1)}$  and  $P_{k|T}^{s(1)}$  for  $k =$ 
    $\{1, \dots, T\}$  using Algorithm 2.
3: for  $j = 1, \dots, N-1$  do
4:   for  $k = 1, \dots, T$  do
5:     Compute  $\mathbf{H}^{(j)}$  at  $\hat{x}_{k-1|k-1}, P_{k-1|k-1}, \hat{x}_{k-1|T}^{s(j)}$ , and
        $P_{k-1|T}^{s(j)}$  using Eq. (17).
6:      $[a_{0,k-1} \ A_{k-1}] =$ 
        $\text{PCE}(f(\cdot), \hat{x}_{k-1|T}^{s(j)}, P_{k-1|T}^{s(j)}, \xi, \mathbf{H}^{(j)})$ .
7:     Compute  $\hat{x}_{k|k-1}$  and  $P_{k|k-1}$  using steps 4 and 5 of
       Algorithm 2.
8:     Evaluate  $\mathbf{H}^{(j)}$  at  $\hat{x}_{k|k-1}, P_{k|k-1}, \hat{x}_{k|T}^{s(j)}$ , and  $P_{k|T}^{s(j)}$ .
9:      $[b_{0,k} \ B_k] = \text{PCE}(h(\cdot), \hat{x}_{k|T}^{s(j)}, P_{k|T}^{s(j)}, \xi, \mathbf{H}^{(j)})$ .
10:    Evaluate  $\hat{x}_{k|k}$  and  $P_{k|k}$  following steps 7 and 8 of
       Algorithm 2.
11:   end for
12:   Compute  $\hat{x}_{k|T}^{s(j+1)}$  and  $P_{k|T}^{s(j+1)}$  using steps 10-16 of
       Algorithm 2.
13: end for
14: end function

```

tracking problems. For both problems, we implemented third-order truncated PCE-based estimators (PCKF, PCRTSS, and IPCRTSS) and four-point Gauss-Hermite (GH)-based estimators (GHF, GHRTSS, and IGHRTSS). The GH-based estimators were chosen for comparison primarily because both methods select the evaluation points as the roots of a Hermite polynomial. In particular, the PCE-based estimators choose a subset of points from the product rule.

A. Pendulum tracking problem

In this example, we consider a pendulum tracking problem with the following state space model [2, pp. 117–118]

$$\begin{aligned} x_k &= \begin{bmatrix} x_{1,k-1} + x_{2,k-1} t \\ x_{2,k-1} - g \sin(x_{1,k-1}) t \end{bmatrix} + \eta_{k-1}, \\ y_k &= \sin(x_{1,k}) + \nu_k, \end{aligned}$$

where $x_{1,k-1}$ is the pendulum angle (in rad), $x_{2,k-1}$ is the pendulum angle rate (in rad/s) at time $k-1$, and t is the sampling time. The process noise $\eta_{k-1} \sim \mathcal{N}(0, Q)$ and measurement noise $\nu_k \sim \mathcal{N}(0, 0.1)$, where

$$Q = q_c \begin{bmatrix} \frac{t^3}{2} & \frac{t^2}{2} \\ \frac{t^2}{2} & t \end{bmatrix},$$

and q_c is the process noise intensity. The following parameters are used for the simulation: $g = 9.81 \text{ m/s}^2$, $t = 0.01 \text{ s}$, and $q_c = 0.01 \text{ rad}^2/\text{s}^3$. The estimation process spans five seconds. The initial truth of the state is set as $x_0 = [1.5 \ 0]^\top$. Here, we have chosen an initial estimate of the state and covariance as $\hat{x}_{0|0} = 0_{2 \times 1}$ and $P_{0|0} = I_{2 \times 2}$, respectively.

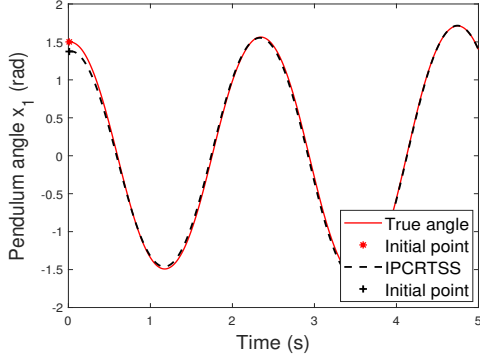


Fig. 2 The pendulum angle truth trajectory and its estimated result with the proposed iterated polynomial chaos RTS smoother for a single run.

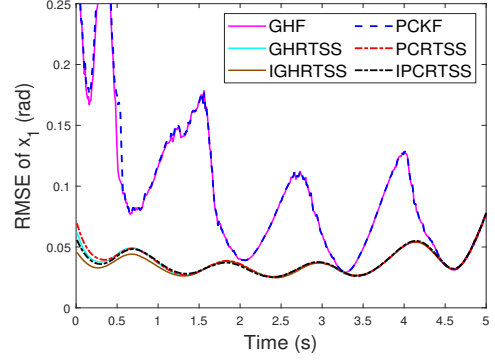
TABLE I Average RMSE values for the pendulum angle (x_1) and angle rate (x_2) obtained from various estimators over 100 MC runs.

Estimators	x_1 (rad)	x_2 (rad/s)
GHF	0.0992	0.2086
GHRSTS	0.0378	0.0999
IGHRTSS	0.0365	0.0945
PCKF	0.1009	0.2062
PCRTSS	0.0381	0.1008
IPCRTSS	0.0377	0.0987

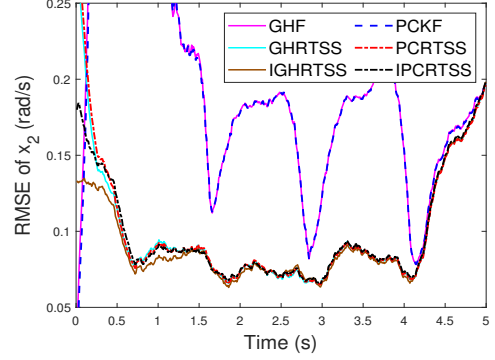
The state is estimated by the third-order chaos expansion-based smoothers (PCKF, PCRTSS, and IPCRTSS) and four-point GH-based estimators (GHF, GHRSTS, and IGHRTSS). Five number of iterations (N) are used for the iterative smoothers. The required number of points for the GH-based estimators is $4^2 = 16$, whereas the PCE-based estimators need $\binom{2+3}{3} = 10$ points. Fig. 2 shows the true pendulum angle along with the estimated result of the IPCRTS smoother for a single run. The performance of the estimators is compared in terms of the root mean squared error (RMSE) obtained from 100 Monte Carlo (MC) runs and plotted in Fig. 3. From Fig. 3, it can be seen that the PCE-based estimators attain a similar RMSE to the GH-based estimators but with less computational burden, as they use fewer points. The GHRSTS and proposed PCRTSS demonstrate better estimation accuracy compared to GHF and PCKF, whereas iterative smoothers provide almost similar (slightly better) estimation accuracy than the respective smoothers. To better visualize the iterative smoothers' performances, we provide the average RMSE value in Table I. From Table I, we see that the utilization of iterative smoothers further enhances estimation accuracy.

B. Aircraft tracking problem

Here, we consider an air-traffic control scenario [10], where an aircraft performs a maneuver in a two-dimensional space with an unknown time-varying turn rate. The dynamics of the maneuvering aircraft in the discrete-time domain can be



(a) RMSE of the Pendulum angle



(b) RMSE of the pendulum angle rate

Fig. 3 The RMSE values for the pendulum angle and its rate, obtained from 100 MC runs, are calculated by various estimators in the pendulum tracking problem.

expressed as

$$x_k = \begin{bmatrix} 1 & \frac{\sin \omega t}{\omega} & 0 & -\frac{1 - \cos \omega t}{\omega} & 0 \\ 0 & \cos \omega t & 0 & -\sin \omega t & 0 \\ 0 & \frac{1 - \cos \omega t}{\omega} & 1 & \frac{\sin \omega t}{\omega} & 0 \\ 0 & \sin \omega t & 0 & \cos \omega t & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + \eta_{k-1},$$

where the state of the aircraft, $x_k = [x_{1,k} \ \dot{x}_{1,k} \ x_{2,k} \ \dot{x}_{2,k} \ \omega_k]^T$, $(x_{1,k}, x_{2,k})$ and $(\dot{x}_{1,k}, \dot{x}_{2,k})$ represent the position and velocity of the aircraft in the x and y directions, respectively; t is the sampling time. The process noise $\eta_{k-1} \sim \mathcal{N}(0, Q)$ with $Q = \text{diag}(q_1\beta, q_1\beta, q_2t, q_2t)$, where

$$\beta = \begin{bmatrix} \frac{t^3}{2} & \frac{t^2}{2} \\ \frac{t^2}{2} & t \end{bmatrix},$$

with the process noise intensities parameters q_1 and q_2 . A radar at the origin of the xy -plane measures the range and bearing of the target, and the measurement equation can be expressed as

$$y_k = \begin{bmatrix} \sqrt{x_{1,k}^2 + x_{2,k}^2} \\ \tan^{-1} \left(\frac{x_{2,k}}{x_{1,k}} \right) \end{bmatrix} + \nu_k,$$

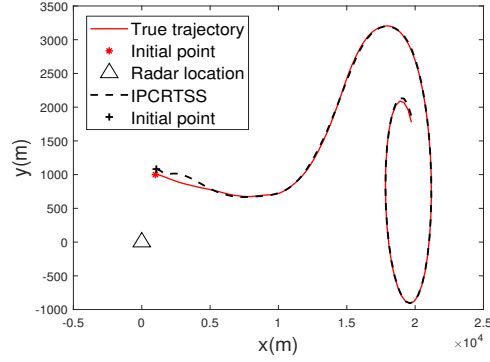


Fig. 4 The truth target trajectory and estimated trajectory of the proposed iterated PCRTS smoother for a single representative run.

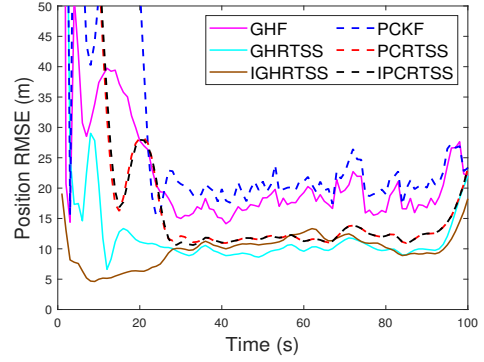
where the measurement noise $\nu_k \sim \mathcal{N}(0, R)$ with $R = \text{diag}(\sigma_1^2, \sigma_2^2)$. The following parameters are used in the simulation: $t = 1$ s, $q_1 = 0.1 \text{ m}^2/\text{s}^3$, $q_2 = 1.75 \times 10^{-4} \text{ s}^{-3}$, $\sigma_1 = 10$ m, $\sigma_2 = \sqrt{10} \times 10^{-3}$ rad. Estimation is performed for 100 seconds. The truth of the target is taken as $x_0 = [1000 \text{ m} \ 300 \text{ m/s} \ 1000 \text{ m} \ 0 \text{ m/s} \ -3^\circ/\text{s}]^\top$. We initialized the estimators with the initial posterior state estimate $\hat{x}_{0|0} = [1200 \text{ m} \ 305 \text{ m/s} \ 1100 \text{ m} \ 4 \text{ m/s} \ -3^\circ/\text{s}]^\top$, and the initial error covariance, $P_{0|0} = \text{diag}(100 \text{ m}^2, 10 \text{ m}^2/\text{s}^2, 100 \text{ m}^2, 10 \text{ m}^2/\text{s}^2, 0.1 \text{ rad}^2/\text{s}^2)$.

We implemented third-order truncated PCE-based estimators (PCKF, PCRTS, and IPCRTS with $N = 5$) and four-point GH-based estimators (GHF, GHRTS, and IGHRTS with $N = 5$) to track the aircraft trajectory. In this example, the number of points required for the GH and PCE-based estimators are $4^5 = 1024$ and $\binom{5+3}{3} = 56$, respectively. Fig. 4 depicts the radar located at the origin, the target's true trajectory, and the estimated trajectory obtained by iterative PCRTS smoother in a single run. It can be seen from the figure that the proposed IPCRTSS successfully tracks the trajectory of the target.

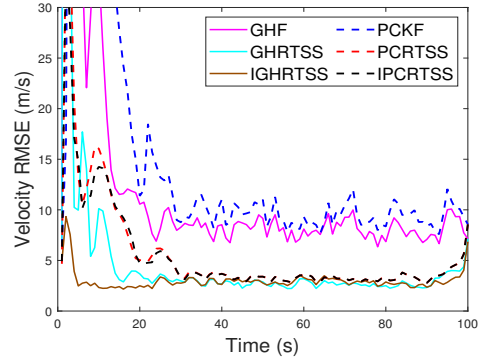
The performance of the estimators is compared in terms of the position and velocity RMSE. We compute the position RMSE at k -th time step from the M MC runs as follows:

$$\text{Pos}_k = \sqrt{\frac{1}{M} \sum_{i=1}^M (x_{1,k}^i - \hat{x}_{1,k}^i)^2 + (x_{2,k}^i - \hat{x}_{2,k}^i)^2},$$

where $x_{1,k}^i$ represents the truth position state at the k -th time-step of the i -th MC run, and $\hat{x}_{1,k}^i$ is its estimate. Similar to the position RMSE, the velocity RMSE can also be computed. The position and velocity RMSE of the different estimators obtained from 100 MC runs are plotted in Fig. 5. The figure shows that the chaos-based estimators achieve a similar accuracy (slightly higher RMSE) as the GH-based estimators at a lower computational cost. The proposed PCRTS smoothers attain lower RMSEs than the PCKF, as expected. The RMSEs are further reduced by using iterative smoother.



(a) Position RMSE



(b) Velocity RMSE

Fig. 5 The position and velocity RMSE of the various estimators for the aircraft tracking problem, obtained from 100 MC runs.

V. CONCLUSION

In this article, we have developed two novel smoothing algorithms, a Gaussian approximation-based polynomial-chaos RTS (PCRTS) smoother and its iterative variant, the IPCRTS smoother. These algorithms use the polynomial chaos expansion to approximate the nonlinear functions in the state space model. The associated integrals of the algorithm are approximately evaluated by fitting the polynomial to the nonlinear functions using a set of collocation points. The performance of the methods was illustrated in the simulated pendulum and aircraft tracking experiments.

REFERENCES

- [1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2002.
- [2] S. Särkkä and L. Svensson, *Bayesian Filtering and Smoothing*, 2nd ed. Cambridge University Press, 2023.
- [3] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [4] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, 2006.
- [5] N. Nadarajah, R. Tharmarasa, M. McDonald, and T. Kirubarajan, "IMM forward filtering and backward smoothing for maneuvering target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2673–2678, 2012.

- [6] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.
- [7] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.
- [8] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, 2000.
- [9] 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 Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.
- [10] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.
- [11] S. Särkkä, "Unscented Rauch–Tung–Striebel smoother," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, 2008.
- [12] N. Chopin and O. Papaspiliopoulos, *An introduction to sequential Monte Carlo*. Springer, 2020.
- [13] C. T. Leondes, J. B. Peller, and E. B. Stear, "Nonlinear smoothing theory," *IEEE Transactions on Systems Science and Cybernetics*, vol. 6, no. 1, pp. 63–71, 1970.
- [14] I. Arasaratnam and S. Haykin, "Cubature Kalman smoothers," *Automatica*, vol. 47, no. 10, pp. 2245–2250, 2011.
- [15] S. Särkkä and J. Hartikainen, "On Gaussian optimal smoothing of nonlinear state space models," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1938–1941, 2010.
- [16] J. Sarmavuori and S. Särkkä, "Fourier–Hermite Rauch–Tung–Striebel smoother," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2109–2113.
- [17] Á. F. García-Fernández, L. Svensson, and S. Särkkä, "Iterated posterior linearization smoother," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 2056–2063, 2017.
- [18] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938.
- [19] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002.
- [20] P. Dutta and R. Bhattacharya, "Nonlinear estimation of hypersonic state trajectories in Bayesian framework with polynomial chaos," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1765–1778, 2010.
- [21] J. Li and D. Xiu, "A generalized polynomial chaos based ensemble Kalman filter with high accuracy," *Journal of Computational Physics*, vol. 228, no. 15, pp. 5454–5469, 2009.
- [22] Y. Xu, L. Mili, and J. Zhao, "A novel polynomial-chaos-based Kalman filter," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 9–13, 2019.
- [23] K. Kumar, R. K. Tiwari, S. Bhaumik, and P. Date, "Polynomial chaos Kalman filter for target tracking applications," *IET Radar, Sonar & Navigation*, vol. 17, no. 2, pp. 247–260, 2023.
- [24] J. Sarmavuori and S. Särkkä, "Fourier–Hermite Kalman filter," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1511–1515, 2011.