

# NONLINEAR FILTERING WITH INTRUSIVE POLYNOMIAL CHAOS FOR SATELLITE UNCERTAINTY QUANTIFICATION

Z McLaughlin\*, Brandon A. Jones<sup>†</sup>, and Renato Zanetti<sup>†</sup>

Polynomial chaos expansions are capable of accurately and efficiently modeling uncertainty even in cases with highly nonlinear dynamics or high state uncertainty. For orbits problems concerning orbit determination and space situational awareness, the ability to achieve accurate state estimates using minimal measurement data is needed to reduce demand on ground based sensors and improve knowledge of debris and untracked objects. This work presents a filter leveraging intrusive Polynomial Chaos Expansions (PCEs) to track an object's probability distribution even as it becomes non-Gaussian in environments with scarce measurements, high a priori uncertainty, and nonlinear dynamics. The following discussion covers background on PCEs and polynomial filtering and comparison to sampling methods and popular filters used in astrodynamics.

## INTRODUCTION

Propagating and quantifying the uncertainty associated with a Anthropogenic Space Object (ASO) is a capability necessary for many commercial space organizations, government entities, and scientists to facilitate ASO tracking, orbit maintenance, and the collection of in-situ science data on satellite missions. Space Situational Awareness (SSA), a field dedicated to cataloging and maintaining custody of ASOs, constantly sees new research in Orbit Determination (OD), Uncertainty Quantification (UQ), and nonlinear filtering methods to improve the amount of knowledge gained from a certain measurement of an ASO using a ground or space-based sensor. As the number of ASOs grows at a rate far outpacing the development of new ground-based sensors, better OD technologies can help negate this disparity.

Due to the highly uncertain a priori Probability Density Functions (PDFs) often dealt with in OD problems, a diverse set of filtering and sampling techniques have been applied to the orbit problem in the literature. Traditional filtering methods, such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF), rely on linear updates, meaning that non-Gaussian information contained in the measurement is not conserved during the update.<sup>1</sup> More modern, accuracy-focused filtering and propagation methods are usually at least quasi-sampling based, where either a set of samples or a mixture of Gaussian distributions may be propagated to the desired time to recover an approximate probability distribution. Gaussian Mixture Models (GMMs) and particle filters are both popular choices in the literature for robust filtering methods.<sup>2-5</sup> GMMs seek to represent a non-Gaussian PDF through a summation of smaller, weighted Gaussian mixands, while particle

---

\*Graduate Research Assistant, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712

<sup>†</sup>Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX, 78712

filters rely on empirical sampling to produce an update robust to nonlinearities.<sup>6-8</sup> Though these methods see widespread implementation, they suffer from some significant drawbacks. Notably for sampling-based methods, the computation time required for high-accuracy evaluations can become high. For composite methods like those using Gaussian mixtures, the same issue may be encountered when more accuracy is required, since this necessitates further splitting the number of components tracked by the filter.

Nonlinearities in the orbits problem (as well as in other filtering problems) arise from two areas; the propagation of the state dynamics and in the collection of a measurement. Because the state dynamics for the orbits problem are nonlinear, any extended propagation will result in a notably non-Gaussian distribution. The types of measurements often used for OD problems include range rate, range, doppler, or angle measurements, such as right ascension and declination, meaning that updates that reduce the associated uncertainty to a Gaussian PDF immediately lose information due to nonlinearity in the measurement unaccounted for by a linear filter update. This work seeks to use polynomial chaos to develop a filter leveraging nonlinear updates to generate an a posteriori state PDF using non-Gaussian information.

Polynomial Chaos Expansions (PCEs) provide an opportunity to approximate higher moments of probability distributions by projecting a stochastic solution onto an orthogonal polynomial basis as a function of random inputs. PCEs show promise on the fronts of both accuracy and computational efficiency, with two broad categories of solutions in non-intrusive PCEs (niPCEs) and intrusive PCEs (iPCEs). Over the last decade, polynomial chaos has started to see use in the field of orbital mechanics for uncertainty propagation, quantification, and its applications (such as for conjunction assessment), with recent contributions being most concentrated in uncertainty quantification.<sup>9-11</sup> Most relevant to this work, Jones et al. began a series of papers that considered the use of non-intrusive PCEs (niPCEs) for these uses, which demonstrated a marked improvement in efficiency in time with an acceptable loss of accuracy compared to Monte Carlo based sampling methods.<sup>12-14</sup> Filtering techniques leveraging niPCEs have also been attempted that use PCEs to perform the time update in the filter, but finish without utilizing the PCE to complete a nonlinear measurement update method.<sup>15,16</sup> These previous forays into the topic each consider niPCEs when approaching the problem, yet it has been marked by researchers in the past that intrusive methods could hold an advantage in convergence and computational efficiency over non-intrusive methodologies.<sup>17,18</sup>

This paper presents an approach to recursive estimation when the stochastic state of the object of interest is represented using intrusive polynomial chaos. First, the authors present an overview of polynomial chaos, including discussion on non-intrusive and intrusive methodologies, the Galerkin projection and its usefulness for intrusive polynomial chaos. Later sections follow this with discussion of the derivation of the intrusive polynomial chaos filter (iPCF), including the development of the time and measurement updates as well as general implementation considerations. Finally, this paper includes results detailing iPCE uncertainty propagation and iPCF performance for both a sun-synchronous Low Earth Orbit (LEO) case as well as a Molniya orbit case, both of which utilize range measurements from simulated geosynchronous spacecraft.

## **POLYNOMIAL CHAOS**

Polynomial chaos methods can be used to model stochastic systems to a high degree of accuracy. A PCE projects the system's solution onto an orthogonal polynomial basis as a function of random inputs. These random inputs share the dimension of the stochastic system — if the orbital motion problem is considered with uncertainty in each Cartesian element, there are 6 random inputs to

the system, and so on. These orthogonal polynomials, Hermite polynomials for Gaussian random variables, are then combined with a set of coefficients to fully describe the estimated solution. Traditionally, this is expressed as a summation of the products of the evaluated polynomials and the coefficients, which can be seen represented mathematically via

$$\hat{\mathbf{X}}(t, \boldsymbol{\xi}) = \sum_{k=0}^P \mathbf{c}_k(t) \psi_k(\boldsymbol{\xi}), \quad (1)$$

where the variables  $\hat{\mathbf{X}}(t, \boldsymbol{\xi})$ ,  $\mathbf{c}_k(t)$ , and  $\psi_k(\boldsymbol{\xi})$  denote the estimated solution given by the PCE, the  $k^{\text{th}}$  coefficient of the PCE, and the  $k^{\text{th}}$  basis function, a product of univariate Hermite polynomials evaluated at a random input vector  $\boldsymbol{\xi}$ .<sup>19,20</sup> In this work, independent and identically distributed Gaussian inputs were used for polynomial generation, though there is no assumption which restricts implementation to only this case, as it is also possible for alternative distributions and non-identically distributed variables.<sup>20,21</sup> The stochastic dimension,  $d$ , combined with the maximum degree of the system,  $p$ , dictates the number of terms

$$P := \frac{(p+d)!}{p!d!}. \quad (2)$$

Once this information is known from the problem selection, the basis functions and coefficients together describe the PCE in its entirety, where the polynomial basis may be generated according to the multi index of the expansion; recall the indexing variable  $\mathbf{k}$  in equation 1 — this variable denotes one scalar index for the expansion that actually corresponds to some multi index for the polynomial basis  $\psi$  such that

$$\psi_{\mathbf{k}}(\boldsymbol{\xi}) = \psi_{k,n_1}(\xi_1) \psi_{k,n_2}(\xi_2) \cdots \psi_{k,n_d}(\xi_d) \quad \mathbf{k} \in \mathbb{N}_0^P, \quad \forall \mathbf{n} \ni \sum_{i=1}^d n_i \leq p, \quad 0 < n_i \leq d. \quad (3)$$

Here,  $\boldsymbol{\xi}$  denotes the total random input vector for a given case,  $d$  is the system's stochastic dimension, and each polynomial has an order  $n_i$ , where the sum of the total order for the multi index cannot exceed the degree of the expansion,  $p$ . By enumerating all possible combinations, the entire PCE basis can be furnished.

As the degree,  $p$ , of the polynomial expansion increases, more information is encoded into the coefficients. The coefficients of the expansion may be used to approximate the statistical moments of the system, with  $c_0$  corresponding to the mean, and where information from the later coefficients may be retrieved to calculate the covariance and other moments. The mean can be written as

$$\mu_f(t) \triangleq \mathbb{E} \left[ \hat{\mathbf{X}}(t, \boldsymbol{\xi}) \right] = \mathbb{E} \left[ \sum_{k=0}^P \mathbf{c}_k(t) \psi_k(\boldsymbol{\xi}) \right] = \hat{f}_0(t), \quad (4)$$

or simply as the  $0^{\text{th}}$  coefficient(s) in the expansion. The covariance and cross-covariance calculated directly from the coefficients are

$$P_{ij} = \sum_{k=0}^P c_{i,k} c_{j,k} \quad i, j \in [0, d], \quad (5)$$

and

$$P_{ij} = \sum_{k=0}^P c_{i,k} b_{j,k} \quad i \in [0, d_c] \quad j \in [0, d_b], \quad (6)$$

where  $P_{ij}$  is an element of  $\mathbf{P}$ , a matrix of shape  $d_c$  by  $d_b$  in the case of cross-covariance computation and a square matrix of shape  $d$  for a covariance matrix. Terms  $c_i$  and  $b_j$  represent the  $i^{\text{th}}$  or  $j^{\text{th}}$  coefficients list for the corresponding PCE in the PCE vector of  $\hat{\mathbf{X}}(t, \xi)$  (and for some other expansion, such as  $\hat{\mathbf{Y}}(t, \eta)$ , for  $b_j$ ). As an alternative to analytic calculation, sampling the PCE can provide empirical data that may be used to estimate probabilistic moments.

Non-intrusive techniques in the literature have considered the use of sampling-based methods to solve for the PCE coefficients at the time(s) of interest, often attempting to optimize for coefficient sparsity or other parameters using specific techniques.<sup>22–24</sup> For niPCEs, a set of random samples varied according to the initial probability distribution is propagated to the final time. At this point, the coefficients may be solved for using these samples to perform the regression. These methods require propagation of far fewer samples than what is necessary for Monte Carlo techniques, and offer inexpensive sampling once the final coefficients are known. Intrusive techniques have seen less frequent implementation in the literature for the orbit problem, which is likely due to the greater effort required on behalf of the user to implement iPCEs. For this work, the Galerkin projection is used to manipulate the individual PCEs comprising the entire stochastic system (with modeling of a priori, measurement noise, and process noise statistics). Through the Galerkin projection, each intermediate calculation in the equations of motion that depends on a stochastic input may also be described through the use of a PCE. The PCE coefficients for each state element derivative may then be integrated numerically, producing the PCE coefficients at any desired time, thus creating a solution for the expansion and a way to cheaply sample from a distribution using pseudo-randomly generated numbers without need for further propagation.<sup>19,25</sup> Specifically, the goal of an iPCE approach is to derive the ODEs

$$\dot{c}_i = \mathbf{f}(i, \mathbf{c}(t)) \quad \forall i \in [0, P] \quad (7)$$

that allow for numeric integration of the coefficients, where some initial PCE  $\hat{\mathbf{X}}(t, \Psi(\mathbf{k}))$  having the coefficients  $\mathbf{c}$ . This means that to perform an intrusive technique, the goal is to derive the function  $f$  for a given stochastic system. By developing arithmetic techniques able to work directly on the PCEs and coefficients, coefficient derivatives may be calculated using the same form of a given ODE with modified operator definitions.

To complete this process for the two-body problem (and to create a software toolkit extensible to other cases), algorithms presented in 25 and 26 were used to create the necessary utilities. This book outlines the necessary computations to formulate the solutions to nonlinear PCE operations. Beginning with Eq. 8, this section presents algorithms defining common nonlinear Galerkin operations to provide some understanding of what taking the Galerkin projection entails for nonlinear systems. Consider the case of the product of two random variables,  $u$  and  $v$ , both dependent on the same vector of random inputs,  $\xi$ , where

$$u(\xi) = \sum_{k=0}^P u_k \Psi_k(\xi), \quad \text{and} \quad v(\xi) = \sum_{k=0}^P v_k \Psi_k(\xi). \quad (8)$$

Their product, the random variable  $w$ , using a PCE is

$$w(\boldsymbol{\xi}) = \sum_{i=0}^P \sum_{j=0}^P u_i v_j \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}). \quad (9)$$

Note that the form of Eq. 9 does not match that of Eq. 1, which is the final desired formulation. Here, reliance on the orthogonal polynomials and their expected values becomes crucial in higher dimensions. We desire a polynomial chaos expansion over the basis  $\Psi_k(\boldsymbol{\xi})$  with coefficients  $w_k$ , where  $k \in [0, P]$ . To achieve this, we can project Eq. 9, containing expansions with bases  $\Psi_i(\boldsymbol{\xi})$  and  $\Psi_j(\boldsymbol{\xi})$ , with  $i \in [0, P]$  and  $j \in [0, P]$ , onto the basis of  $w(\boldsymbol{\xi})$ . This operation leads to the development of the three dimensional tensor,

$$C_{ijk} \equiv \frac{\langle \Psi_i \Psi_j \Psi_k \rangle}{\langle \Psi_k \Psi_k \rangle}, \quad (10)$$

referred to in this work as the Galerkin Product Tensor.<sup>25</sup> Here, the three dimensional tensor with indices  $(i, j, k)$  is shown to be the inner product (or expected value of the product of) polynomials  $i$ ,  $j$ , and  $k$  for the orthogonal set divided by the expected norm of polynomial  $k$ . Note that this tensor is only a function of the expectation of the orthogonal polynomials, and thus may be pre-computed and stored. Upon projecting Eq. 9 onto the orthogonal basis  $\Psi_k$ , then the product of two PCEs may be computed as

$$w_k = \frac{\langle w, \Psi_k \rangle}{\langle \Psi_k^2 \rangle} = \sum_{i=0}^P \sum_{j=0}^P u_i v_j C_{ijk}, \quad (11)$$

meaning the tensor is used to aid in the computation of the PCE coefficients of  $w$ ,  $w_k$ , for  $k \in [0 \dots P]$ . Equation 11 shows how this tensor is used to perform operations to aid in the computation of the PCE coefficients of  $w$ ,  $w_k$ , for  $k \in [0 \dots P]$ . After multiplication, we may consider inversion, expressed as

$$u^{-1}(\boldsymbol{\xi}) = \frac{1}{u(\boldsymbol{\xi})} = \left( \sum_{k=0}^P u_k \Psi_k(\boldsymbol{\xi}) \right)^{-1}, \quad (12)$$

and

$$u^{-1}(\boldsymbol{\xi})u(\boldsymbol{\xi}) = 1, \quad (13)$$

which describe the defined PCE for the inverse of random variable  $u$ . Eq. 13 may then be expressed in the form of a linear system after appropriate substitutions, where the vector of PCE coefficients for the inverted variable may be solved for through the system of linear equations as

$$\begin{pmatrix} \sum_{j=0}^P C_{j00}u_j & \cdots & \sum_{j=0}^P C_{jP0}u_j \\ \vdots & \ddots & \vdots \\ \sum_{j=0}^P C_{j0P}u_j & \cdots & \sum_{j=0}^P C_{jPP}u_j \end{pmatrix} \begin{pmatrix} u_0^{-1} \\ \vdots \\ u_P^{-1} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 0 \end{pmatrix}. \quad (14)$$

Through a similar set of equations, the division of random variables  $u$  and  $v$  may be written as

$$w = \frac{u}{v}, \quad (15)$$

and

$$\begin{pmatrix} \sum_{j=0}^P C_{j00}v_j & \cdots & \sum_{j=0}^P C_{jP0}v_j \\ \vdots & \ddots & \vdots \\ \sum_{j=0}^P C_{j0P}v_j & \cdots & \sum_{j=0}^P C_{jPP}v_j \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_P \end{pmatrix} = \begin{pmatrix} u_0 \\ \vdots \\ u_P \end{pmatrix}, \quad (16)$$

where the quotient's coefficients, again denoted as  $w_k$ , for  $k \in [0 \dots P]$ , may be found. The final Galerkin projection nonlinear operation utilized in this work is that for the square root. This operation is different in that it may not be solved analytically; instead a system of nonlinear equations must be composed, again using the Galerkin product matrix, taking the form

$$\sqrt{u}(\boldsymbol{\xi})\sqrt{u}(\boldsymbol{\xi}) = u(\boldsymbol{\xi}), \quad (17)$$

and

$$\begin{pmatrix} \sum_{j=0}^P C_{j00}\sqrt{u}_j & \dots & \sum_{j=0}^P C_{jP0}\sqrt{u}_j \\ \vdots & \ddots & \vdots \\ \sum_{j=0}^P C_{j0P}\sqrt{u}_j & \dots & \sum_{j=0}^P C_{jPP}\sqrt{u}_j \end{pmatrix} \begin{pmatrix} \sqrt{u}_0 \\ \vdots \\ \sqrt{u}_P \end{pmatrix} = \begin{pmatrix} u_0 \\ \vdots \\ u_P \end{pmatrix}, \quad (18)$$

for the polynomial expansion  $u(\boldsymbol{\xi})$ . These equations differ from the inversion and division processes in that the coefficient matrix and the vector on the left-hand-side of the equation both contain the same unknown, the coefficients  $\sqrt{u}_c$ . This process may be solved using an iterative numerical method such as Newton-Raphson by using an initial guess for  $\sqrt{u}_0$  as  $\pm\sqrt{u_0}$  (note the subscript as being inside the root for this case). This process is essentially using the knowledge of the first coefficient as the mean value and solves the system by the knowledge that the square root of the mean of the input expansion should be approximately equal to the mean of the (square rooted) output expansion.

These operations have been implemented in both python and c/c++ for use in PCE operations during the time update and measurement update of the developed polynomial chaos filter. By using a Galerkin Tensor generated based on the polynomial basis for a specific set of random inputs, each of these operations is immediately extensible to a problem corresponding to that generated tensor. This means that, once these tools have been developed, they may be extended to any system of differential equations. Here it is prudent to note that the Galerkin Tensor,  $C_{ijk}$ , is sparse, and that the majority of values in the tensor will always be zero; this means that careful implementation and indexing techniques can lead to significant computational savings in Galerkin operations.

## POLYNOMIAL FILTERING

By understanding the way the Galerkin projection can be used to manipulate existing PCEs, a filtering algorithm may be developed that utilizes them to perform updates to an object's PDF. The iPCE-based filtering algorithm presented here performs nonlinear updates using the polynomial coefficients to access information regarding the higher-order statistical moments of a tracked object. Previous work by Servadio et al.<sup>27</sup> demonstrated the use of such polynomial-based filtering for Taylor polynomials (based on monomials), which provided further opportunities for investigation using other orthogonal polynomial bases (such as those used in a PCE).<sup>27</sup>

While in earlier work, Taylor polynomials were used in the filtering methods to promote generality, they are not optimal when information on an object's probability distribution is known. For example, an expansion of monomials required the use of Isserlis' moment of a Gaussian distribution to compute moments of a quantity of interest, whereas basis functions comprised of Hermite polynomials directly correspond to a Gaussian distribution. This makes PCEs a good choice for further work using a polynomial-based filtering method. Rather than relying on Isserlis' moment, a PCE's first order moments may be computed using the coefficients via Eq. 4, and the second order moments may be computed analytically using the coefficients via Eq. 5 and Eq. 6 for the cross covariance, as discussed in earlier sections.

This paper follows this previous work developed in 27 by applying the quadratic update scheme introduced there to intrusive polynomial chaos using the Galerkin projection; this required implementation of an intrusive Galerkin propagation scheme and the ability to index the resultant PCE coefficients and Galerkin tensor to perform the higher order measurement updates — these utilities may be constructed using the guidelines presented in the previous section.

### Assumptions and Conventions

Assumptions made in the probabilistic models include those pertaining to the PCE coefficients and what they represent, as well as the ways that operations may be performed between random quantities. The measurement error,  $\epsilon$ , is additive to the state,

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, t) + \epsilon_k, \quad (19)$$

after it has been passed through the measurement function,  $\mathbf{h}(\mathbf{x}, t)$ . Similarly, the process noise is additive to the acceleration of the dynamical system,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \nu_{k-1} \quad (20)$$

and both it and the measurement noise are assumed to be zero mean and uncorrelated for all times. Additionally, the process noise and the measurement noise are both assumed Gaussian, and for a PCE modeling a Gaussian distribution, all coefficients with a basis function higher than one is zero. Other implementations may be used for the measurement noise and process noise PDFs, but they are not considered here. The initial prior state distribution for all test cases was a Gaussian PDF. Each of these conventions are followed for the PCE implementation of the filter. For example, the measurement polynomial is additive to the state polynomial after it has been used in the measurement model function  $\mathbf{h}(\mathbf{x}, t)$ . The entire filter operates off of the PCE coefficients list for the prior, the measurement noise, and the process noise, with the polynomial basis used to compute the Galerkin Product Tensor at the start of the simulation (unless it has already been generated and saved) and the multi-index of the PCE, which is used to index through the expansions to correctly combine and manipulate PCEs. Further subsections explain how the filter is initialized, how the time update is used to propagate the PCE coefficients, and how the quadratic update is performed to arrive at a posterior leveraging the measurement and process noise PCEs.

### Filter Initialization

To begin filtering with intrusive polynomial chaos, the relevant PCEs (polynomial basis and coefficients) and Galerkin Product Tensor must be generated. Most of this step can be generalized for a given set of dynamics and measurement dimension; only the initial PCE coefficients must be recalculated if changes to the initial mean, covariance, or other surface-level statistics are made. Governing the polynomial basis and the product tensor are the stochastic dimension of the system, the degree of the expansion and its component parts, and the type of random input (RI) used to model the system.

For this work, the total set of PCEs used in the filter include those describing the prior PDF, the process noise statistics, and the measurement noise statistics, denoted as  $\mathbf{P}(\xi, \nu, \eta)$ , where  $\xi, \nu, \eta$  are three random input vectors with dimensions matching the PDFs for the prior state, the process noise, and the measurement noise, respectively. This paper assumes all three of these random input vectors can be modeled as unit normal distributions, described by  $\mathcal{N}(0, 1)$ .  $\mathbf{x}$  refers here to the PCE

representation of the estimated state, and its dimension throughout this section should be assumed to be a function of the random input vectors included in the parenthesis. The state PDF PCEs can be written as  $\boldsymbol{x}(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta})$  when it is dependent on all three RI vectors, meaning  $\boldsymbol{x}$  is dependent on the prior statistics RIs, the measurement noise statistics RIs, and the process noise RIs.

When the PCE is first generated, each set of expansions is independent from one another, meaning that as stated under the assumptions section, the filter assumes no correlation between the state PDF, the measurement statistics, and the process noise statistics. The list of PCEs can then be split and denoted as  $\boldsymbol{x}(\boldsymbol{\xi})$  for the state,  $\boldsymbol{\theta}(\boldsymbol{\nu})$  for the process noise, and  $\boldsymbol{\epsilon}(\boldsymbol{\eta})$  for the measurement noise. The PCEs for  $\boldsymbol{\theta}(\boldsymbol{\nu})$  and  $\boldsymbol{\epsilon}(\boldsymbol{\eta})$  are restricted to degree one, which implies that the measurement and process PDFs are Gaussian in nature.

Before proceeding with the time update of the filter, a polynomial basis constructed from the univariate Hermite polynomials must be generated and used to create the product tensor for the problem as outlined in the Polynomial Chaos section. After this is complete, Eq. 11 may be followed to populate the tensor elements. Using the multi index, tensor elements can be arranged for convenience. Because the degree of the PCE will double during the quadratic update, for example, it is computationally efficient to move the higher order terms in the tensor to the end such that indexing may be truncated where possible.

### Time Update

The time update of a PCE representation of  $\boldsymbol{x}_k$  uses the intrusive approach described above. Here, the coefficients of the expansion are propagated directly using numerical integration just as the state would be integrated for a deterministic numerical integration problem, only following the arithmetic outlined for the Galerkin projection on the PCE coefficients. The propagated a priori polynomial,

$$\boldsymbol{x}_k^- (\boldsymbol{\xi}, \boldsymbol{\nu}) = \mathbf{f} (\boldsymbol{x}_{k-1} (\boldsymbol{\xi}), \boldsymbol{\theta} (\boldsymbol{\nu})) \quad (21)$$

denoted as  $\boldsymbol{x}_k^- (\boldsymbol{\xi}, \boldsymbol{\nu})$ , indicates its pre-update status using the superscript  $-$ , and shows that its now a function of both the process noise and the prior RIs due to the addition of the process noise PCEs in the differential equations.

### Quadratic Measurement Update

This process is largely replicated from the work already presented in 27, with the only significant differences at the update stage being the type of polynomial (PCEs rather than Taylor expansions). The polynomial estimator is formulated to produce an estimate of some random state vector,  $\boldsymbol{x}$ , using information from another random vector  $\boldsymbol{y}$ , and is defined based on a family of quadratic estimators,  $\mathbf{g}(\boldsymbol{y})$ ,

$$\mathbf{g}(\boldsymbol{y}) = \mathbf{a} + \mathbf{K}_1 \boldsymbol{y} + \mathbf{K}_2 \boldsymbol{y}^{[2]}, \quad (22)$$

where  $\boldsymbol{y}^{[2]}$  leads to the squared calculation necessary for the development of the quadratic update. Using the definition in Eq. 30,  $\boldsymbol{y}^{[2]}$  may be formulated for the random quantity of interest, in this case the measurement polynomials, after which the deviations

$$d\boldsymbol{y} = \boldsymbol{y} - \mathbb{E}\{\boldsymbol{y}\} \quad (23)$$

and

$$d\boldsymbol{y}^{\{2\}} = \boldsymbol{y} \otimes \boldsymbol{y} - \mathbb{E}\{\boldsymbol{y} \otimes \boldsymbol{y}\} = \boldsymbol{y}^{[2]} - \mathbb{E}\{\boldsymbol{y}^{[2]}\}, \quad (24)$$

may be defined and used to form a new quadratic estimator family,

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1(\mathbf{y} - \mathbb{E}\{\mathbf{y}\}) + \mathbf{K}_2 \left( \mathbf{y}^{[2]} - \mathbb{E}\left\{ \mathbf{y}^{[2]} \right\} \right), \quad (25)$$

including the random state variable we wish to solve for,  $\mathbf{x}$ . Further work allows for the use of the orthogonality principle to establish the filter gain in terms of the cross-covariance between  $\mathbf{x}$  and  $\mathbf{y}$ , which results in the optimal estimator of

$$\hat{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1^* d\mathbf{y} + \mathbf{K}_2^* d\mathbf{y}^{[2]}, \quad (26)$$

where  $\mathbf{K}_1^*$  and  $\mathbf{K}_2^*$  are optimal estimates for the values  $\mathbf{K}_1$  and  $\mathbf{K}_2$  enumerated below in Eq. 39. For a more complete derivation of this process, see 27.

With the mathematical underpinnings of the estimator defined as well as the techniques established in earlier sections for computing PCE moments, the quadratic update implemented in the filter may be written. The nonlinear measurement update of the filter sees an expansion in the degree of the PCE to double the starting degree through a quadratic update after a measurement,  $\mathbf{y}_k$  is received. After propagation is complete, the propagated a priori is used to compute the measurement polynomial,

$$\mathbf{y}_k(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta}) = \mathbf{h}(\mathbf{x}_k^-(\boldsymbol{\xi}, \boldsymbol{\nu})) + \boldsymbol{\epsilon}(\boldsymbol{\eta}) \quad (27)$$

where  $\mathbf{y}_k(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta})$  is a function of all of the random inputs used for the problem. The function  $\mathbf{h}$  denotes a formulation of the problem's measurement model compatible with the Galerkin projection. The mean values of the propagated a priori and measurement PCEs then may be easily retrieved. For a PCE, the first coefficient of the expansion is always the mean, equivalent to the expectation, written as

$$\bar{\mathbf{x}}_k^- = \mathbb{E}\{\mathbf{x}_k^-\}, \quad (28)$$

and

$$\bar{\mathbf{y}}_k = \mathbb{E}\{\mathbf{y}_k\}. \quad (29)$$

To continue with the quadratic update, squared measurement PCEs must be computed along with the square of the  $k_{th}$  measurement using a modified version the Kronecker product not including duplicate terms,

$$\mathbf{u} \otimes \mathbf{u} \triangleq u_k^2 = u_i u_j, i \in [0, d], j \in [i, d] \forall i \in [0, d] \quad (30)$$

where  $u_k^2$  is an element of the resultant vector  $\mathbf{u}^2$ , and  $u_i$  and  $u_j$  are elements of the input vector  $\mathbf{u}$  having dimension  $d$ . This relationship can be used to express

$$\tilde{\mathbf{y}}_k^{[2]} = \tilde{\mathbf{y}}_k \otimes \tilde{\mathbf{y}}_k, \quad (31)$$

$$\mathbf{y}_k^{[2]} = \mathbf{y}_k \otimes \mathbf{y}_k, \quad (32)$$

and

$$\bar{\mathbf{y}}_k^{[2]} = \mathbb{E}\left\{ \mathbf{y}_k^{[2]} \right\}, \quad (33)$$

the squared values for the measurement, the measurement PCE calculated in the time update, and the squared measurement PCE's mean, respectively. With all of this information, an augmented measurement covariance matrix and a state-measurement cross covariance matrix are needed to

compute the filter's gain. As was the case for the mean value of a PCE, each covariance matrix may be calculated analytically from the PCE's coefficients, but they may also be expressed as

$$\mathbf{P}_{yy} = \mathbb{E} \left\{ (\mathbf{y}_k - \bar{\mathbf{y}}_k) (\mathbf{y}_k - \bar{\mathbf{y}}_k)^T \right\}, \quad (34)$$

$$\mathbf{P}_{yy^{[2]}} = \mathbb{E} \left\{ (\mathbf{y}_{k+1} - \bar{\mathbf{y}}_k) \left( \mathbf{y}_k^{[2]} - \bar{\mathbf{y}}_k^{[2]} \right)^T \right\}, \quad (35)$$

and

$$\mathbf{P}_{y^{[2]}y^{[2]}} = \mathbb{E} \left\{ \left( \mathbf{y}_k^{[2]} - \bar{\mathbf{y}}_k^{[2]} \right) \left( \mathbf{y}_k^{[2]} - \bar{\mathbf{y}}_k^{[2]} \right)^T \right\}, \quad (36)$$

using the expectation operator. After each of these matrices has been calculated, they may be blocked together to produce the full quadratic measurement covariance and the cross covariance, and directly afterward, the filter gain, by the expressions

$$\mathbf{P}_{y\mathcal{Y}} = \left[ \begin{array}{c|c} \mathbf{P}_{yy} & \mathbf{P}_{yy^{[2]}} \\ \hline \mathbf{P}_{yy^{[2]}} & \mathbf{P}_{y^{[2]}y^{[2]}} \end{array} \right], \quad (37)$$

$$\mathbf{P}_{x\mathcal{Y}} = \left[ \begin{array}{cc} \mathbf{P}_{xy} & \mathbf{P}_{xy^{[2]}} \end{array} \right], \quad (38)$$

and

$$\mathbf{K} = \left[ \begin{array}{cc} \mathbf{K}_1 & \mathbf{K}_2 \end{array} \right] = \mathbf{P}_{x\mathcal{Y}} \mathbf{P}_{y\mathcal{Y}}^{-1}. \quad (39)$$

Finally, the a posteriori mean and covariance may be calculated using the measurement polynomials, the measurement, and the gain,  $\mathbf{K}$ , by the expressions

$$\mathbf{x}_k^+ (\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta}) = \mathbf{x}_k^- (\boldsymbol{\xi}, \boldsymbol{\nu}) + \mathbf{K} \left[ \begin{array}{c} \tilde{\mathbf{y}}_k - \mathbf{y}_k^- (\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta}) \\ \tilde{\mathbf{y}}_k^{[2]} - \mathbf{y}_k^{[2]} (\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta}) \end{array} \right], \quad (40)$$

$$\bar{\mathbf{x}}_k^+ = \mathbb{E} \{ \mathbf{x}_k^+ \} = \bar{\mathbf{x}}_k^- + \mathbf{K} \left[ \begin{array}{c} \tilde{\mathbf{y}}_k - \bar{\mathbf{y}}_k \\ \tilde{\mathbf{y}}_k^{[2]} - \bar{\mathbf{y}}_k^{[2]} \end{array} \right], \quad (41)$$

and

$$\mathbf{P}_{xx,k} = \mathbb{E} \left\{ (\mathbf{x}_k^+ - \bar{\mathbf{x}}_k^+) (\mathbf{x}_k^+ - \bar{\mathbf{x}}_k^+)^T \right\}. \quad (42)$$

For now, the iPCF utilizes the quadratic update to calculate the mean and covariance directly, meaning that a Gaussian a posteriori PDF is computed using knowledge of the non-Gaussian prior PDF and the non-Gaussian measurement PDF. The coefficients to begin the next filter step are then computed using a least squares regression to calculate coefficients based on the posterior mean and covariance. These coefficients, describing a degree two expansion dependent only on the prior random variables (but populated only using the posterior mean and covariance), are then used at the start of the next filter step. The a posteriori mean and covariance, denoted as  $\hat{\mathbf{x}}$  and  $\mathbf{P}_{xx}$ , were previously modeled with the random inputs  $\boldsymbol{\xi}$ . This means that using only these random variables, we can sample the a posteriori mean and covariance and replace the coefficients related to the state PDF of the object. Using a matrix,  $\mathbf{H}$ , of evaluations of the polynomials comprising the PCE basis,  $\psi$ , evaluated for the state uncertainty random inputs,  $\boldsymbol{\xi}$ , a least squares regression can be computed by

$$\mathbf{H}\mathbf{C}_{t_f} = \mathbf{Y}_{t_f}, \quad (43)$$

such that

$$\hat{C} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Y} \quad (44)$$

where  $\hat{C}$  denotes the new coefficients and  $(Y)$  are the samples from  $\mathbf{H}$  transformed using the a posteriori mean and covariance, providing a method to quickly reinitialize the second degree PCE describing the state. Because of the polynomial basis ordering performed earlier, assigning these coefficients to the total set of PCEs used in the filter is trivial. After this process is complete, the filter continues with the next time update, and so on for available measurements. Future work will alter alternative methods of PDF continuity, such as truncation of the a posteriori PCE, to explore filter performance without modeling the output PDF as a Gaussian distribution.

## RESULTS

This section presents an orbits test case used to evaluate the performance of the iPCF. Two test cases were considered; a prospective sun-synchronous LEO satellite Molniya orbit. Three orbiting Geosynchronous spacecraft were simulated to generate range measurements to the spacecraft in order to insert measurement variety in the system. These satellites are assumed to be in constant position in an Earth-Centered-Earth-Fixed (ECEF) frame with positions

$$\begin{aligned} \text{Satellite } 1_{\text{ECEF}} &= [ \quad 1.0 \quad 0.0 \quad 0.0 \quad ] \\ \text{Satellite } 2_{\text{ECEF}} &= [ \cos(120^\circ) \quad \sin(120^\circ) \quad 0.0 \quad ], \\ \text{Satellite } 3_{\text{ECEF}} &= [ \cos(240^\circ) \quad \sin(120^\circ) \quad 0.0 \quad ] \end{aligned} \quad (45)$$

that may be converted to an Earth-Centered-Inertial (ECI) frame using a Z-axis transformation based on the simulation time. Based on proximity to the tracked spacecraft, one of the three Geosynchronous spacecraft provides a measurement for each selected time. This section includes two separate results metrics; a Monte Carlo simulation evaluating the fidelity of the iPCE propagation methods as well as a Monte Carlo filter consistency test. The propagation test is examined first and is followed by discussion of filter performance. The initial conditions for both cases are the same; two-body problem dynamics with initial conditions described in Table 1 and statistics as described in Table 2 for both the Molniya and sun-synchronous test cases.

**Table 1. Keplerian orbital elements for the sun synchronous LEO and Molniya orbit test cases**

Case	a (km)	e	i (deg)	$\omega$ (deg)	$\Omega$ (deg)	$\nu$ (deg)
Sun-Synchronous	6945	0.001	97.7	0.0	0.0	0.0
Molniya	26553	0.737	63.4	0.0	270	0.0

**Table 2. Covariance Matrices for the prior state, process noise, and measurement noise PDFs**

Value	Covariance Matrix
$\mathbf{P}_{xx}$	diag[10 km <sup>2</sup> & 10 km <sup>2</sup> & 10 km <sup>2</sup> & 10E-6 $\frac{\text{km}^2}{\text{s}^2}$ & 10E-6 $\frac{\text{km}^2}{\text{s}^2}$ & 10E-6 $\frac{\text{km}^2}{\text{s}^2}$ ]
$\mathbf{Q}$	diag[1E-2, 1E-2] $\frac{\text{km}^2}{\text{s}^4}$
$\mathbf{R}$	diag[1E-2, 1E-2] km <sup>2</sup>

For a test of the intrusive propagator, these initial conditions were propagated 5 orbital periods from the initial time in the sun synchronous case, or  $t_{f,s} = 28800$  seconds. For the Molniya orbit

case, the distribution was propagated for one orbital period, or  $t_{f,m} = 7176$  seconds. At the final time, 10,000 samples were computed from the PCE solution and 10,000 samples were propagated using Monte Carlo techniques. The resultant distribution and mean values are shown in Fig. 1 and Table ???. Fig. 1 shows the samples at  $t_{f,s}$  for the sun synchronous case and  $t_{f,m}$  for the Molniya case in the X, Y, and Z ECI frame components. Tables and show the ECI Cartesian coordinates of the distributions' mean values at the final time, while Tables and and Tables and show the marginal variance and marginal skewness, respectively. Notably, the mean for the Molniya case differs somewhat considerably for all three methods; this is due to the relatively low sample count in addition to the degree of the PCE used. This trial is meant to be illustrative of what kind of capabilities are currently in use in the filter cases, which leverage 2<sup>nd</sup> and 4<sup>th</sup> degree expansions. In addition, it illuminates both graphically and numerically how the iPCE approach can yield gains over traditional filtering. The iPCE method tracks the marginal variances much more closely than the UKF for both cases, implying a better understanding of the PDF. The estimate for skewness was not calculated analytically with the coefficients, though with further implementation it may be done. Instead, it was calculated by sampling the PCE. Each of these results show that for an expansion degree of 3, there are enough coefficients in each to model some non-Gaussian behavior over a long period of time. Visually, the distributions match well, with the iPCE and Monte Carlo methods agreeing in general shape even after long propagation times. The iPCE solution begins to diverge from the MC simulation at this final time, especially in the case of the Molniya orbit, but a higher degree expansion would remedy this. For the filtering problems considered here, this amount of time is generally a far longer gap than what would normally be present in between measurements.

**Table 3. Mean values at  $t_{f,s}$ : sun-sync case**

Value	UT	iPCE	Monte Carlo
$x$ (km)	6.926E3	6.926E3	6.926E3
$y$ (km)	1.221E-1	1.219E-1	5.195E-1
$z$ (km)	-9.031E-1	-9.013E-1	-3.944E0
$\dot{x}$ ( $\frac{\text{km}}{\text{s}}$ )	5.722E-4	5.696E-4	3.877E-3
$\dot{y}$ ( $\frac{\text{km}}{\text{s}}$ )	-1.014E0	-1.014E0	-1.014E0
$\dot{z}$ ( $\frac{\text{km}}{\text{s}}$ )	7.502E0	7.502E0	7.502E0

**Table 4. Mean values at  $t_{f,m}$ : Molniya case**

Value	UT	iPCE	Monte Carlo
$x$ (km)	-1.078E1	-8.750E0	-1.291E1
$y$ (km)	-6.503E3	-6.530E3	-6.520E3
$z$ (km)	-2.152E1	-1.747E1	-2.580E1
$\dot{x}$ ( $\frac{\text{km}}{\text{s}}$ )	4.169E0	4.200E0	4.195E0
$\dot{y}$ ( $\frac{\text{km}}{\text{s}}$ )	-1.449E-2	-1.84E-2	-1.648E-2
$\dot{z}$ ( $\frac{\text{km}}{\text{s}}$ )	8.326E0	8.386E0	8.378E0

**Table 5. Marginal variance at  $t_{f,s}$ : sun-sync case**

Value	UT	iPCE	Monte Carlo
$x(\text{km}^2)$	3.580E2	2.886E2	2.938E2
$y(\text{km}^2)$	2.955E3	2.950E3	2.993E3
$z(\text{km}^2)$	1.611E5	1.608E5	1.633E5
$\dot{x}$ ( $\frac{\text{km}^2}{\text{s}^2}$ )	1.956E-1	1.953E-1	1.982E-1
$\dot{y}$ ( $\frac{\text{km}^2}{\text{s}^2}$ )	1.746E-5	1.598E-5	1.638E-5
$\dot{z}$ ( $\frac{\text{km}^2}{\text{s}^2}$ )	4.181E-4	3.368E-4	3.453E-4

**Table 6. Marginal variance at  $t_{f,m}$ : Molniya case**

Value	UT	iPCE	Monte Carlo
$x(\text{km}^2)$	2.312E6	2.168E6	2.200E6
$y(\text{km}^2)$	5.096E5	3.381E5	3.590E5
$z(\text{km}^2)$	9.221E6	8.644E6	8.772E6
$\dot{x}$ ( $\frac{\text{km}^2}{\text{s}^2}$ )	1.837E-1	9.834E-2	9.731E-2
$\dot{y}$ ( $\frac{\text{km}^2}{\text{s}^2}$ )	6.907E0	5.761E0	5.906E0
$\dot{z}$ ( $\frac{\text{km}^2}{\text{s}^2}$ )	7.323E-1	3.921E-1	3.880E-1

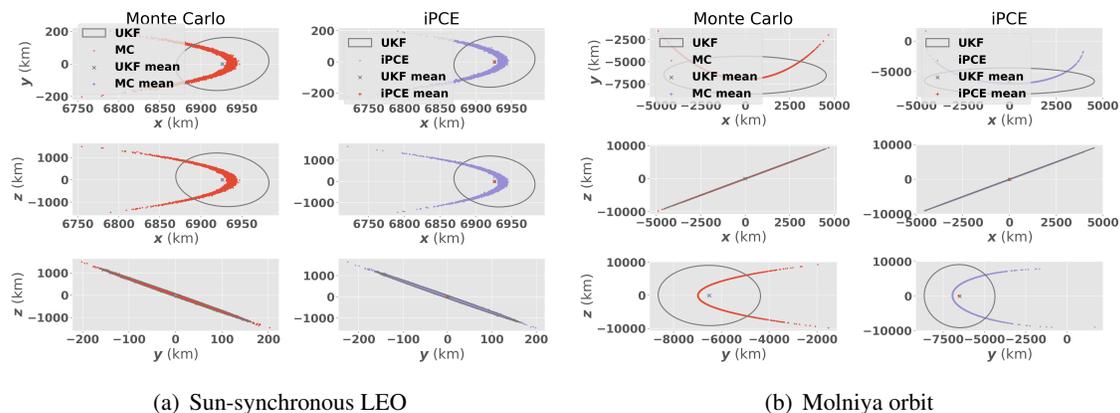


Figure 1. Monte Carlo propagation trial

Table 7. Marginal skewness at  $t_{f,s}$ : sun-sync case

Value	UT	iPCE	Monte Carlo
$x$	—	-2.877	-2.645
$y$	—	-0.033	0.036
$z$	—	0.031	-0.033
$\dot{x}$	—	-0.037	0.027
$\dot{y}$	—	0.671	0.68
$\dot{z}$	—	-2.872	-2.642

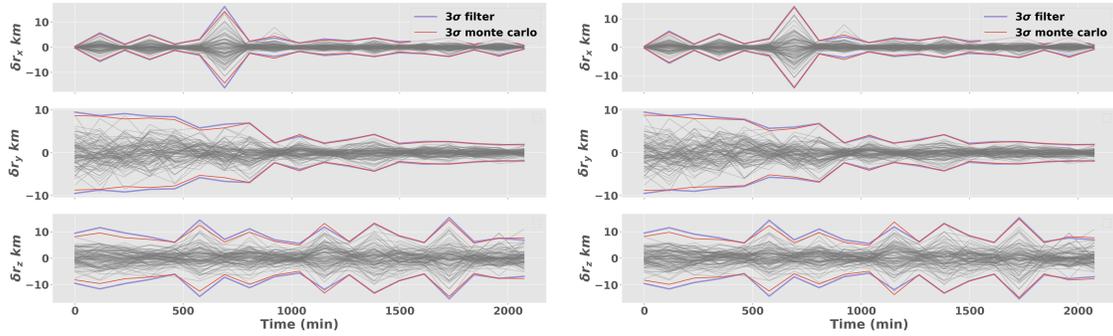
Table 8. Marginal skewness at  $t_{f,m}$ : Molniya case

Value	UT	iPCE	Monte Carlo
$x$	—	-0.073	-0.022
$y$	—	2.795	2.216
$z$	—	-0.073	-0.022
$\dot{x}$	—	-2.957	-1.748
$\dot{y}$	—	0.008	-0.007
$\dot{z}$	—	-2.959	-1.749

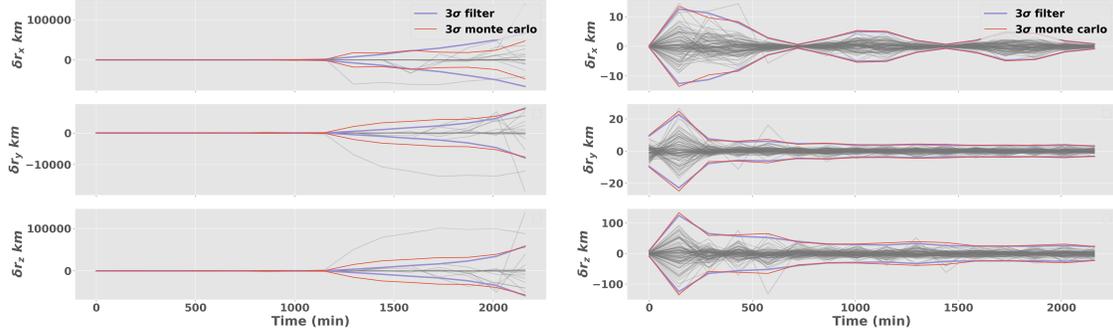
The propagation results show fidelity in the intrusive Galerkin utilities with a close qualitative match to the Monte Carlo propagation. The iPCE uses this algorithm to perform the time update; the remainder of the filtering step is then executed by through the quadratic polynomial update, described earlier in the paper, on the propagated a priori polynomials and the measurement polynomials calculated from them at a given time. Current results from the PCE filter indicate success at handling scarce measurement data in highly uncertain scenarios. For the same LEO sun-synchronous test case and Molniya orbit test case outlined in Tables 1 and 2 using two-body dynamics with process noise additive in acceleration, the iPCE filter remains consistent in all Monte Carlo trials performed, while the baseline UKF test algorithm fails in the case with the lowest measurement density due to an inability to accurately describe the non-Gaussian probability distribution.

Figures 2, 3, 4 show the results of three different 100 trial Monte Carlo simulations performed using the iPCE filter against a UKF baseline. These each show the state error for the Monte Carlo trials along with the  $3\sigma$  boundaries produced by the filter and calculated with the Monte Carlo data. The statistics on the measurement noise, the additive process noise, and the prior state are listed in Table 2. Each trial ended after 15 orbital periods for its respective test case. The measurement frequency is listed in seconds in each figure caption, with the satellite receiving a range measurement from one of 3 simulated geostationary spacecraft at each time step.

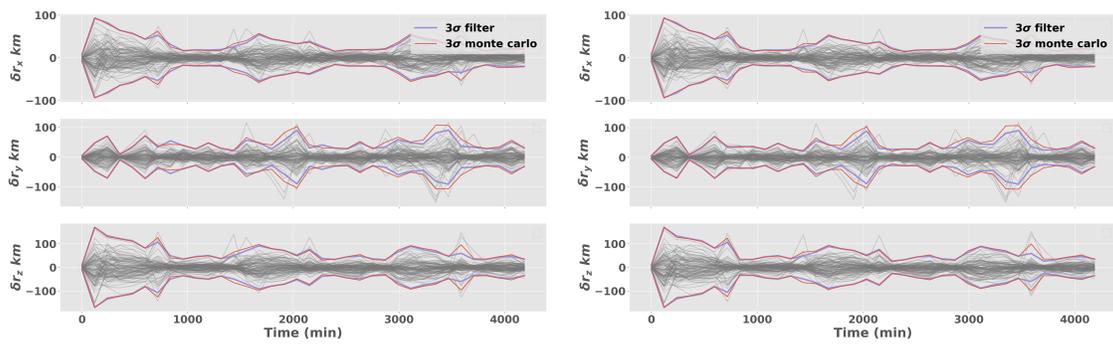
Figure 2 shows a case with measurements every 1.2 periods where the UKF is still able to provide a converging solution and show consistent filter behavior. The iPCE and UKF show similar results here, in part due to the reduction of the posterior to a Gaussian distribution by the current



(a) UKF (b) iPCF  
**Figure 2. 100 run Monte Carlo filter trial: sun synchronous,  $t_{step} = 6911$  seconds**



(a) UKF (b) iPCF  
**Figure 3. 100 run Monte Carlo filter trial: sun synchronous,  $t_{step} = 8639$  seconds**



(a) UKF (b) iPCF  
**Figure 4. 100 run Monte Carlo filter trial: Molniya,  $t_{step} = 7176$  seconds**

version of the iPCF. Figure 3 illustrates what happens when the measurement sparsity is increased significantly; the UKF diverges wildly, with higher uncertainty than the iPCF even before it completely separates from the true state. The iPCF is able to remain a consistent estimator even in this case, showing it's extensibility to more difficult problems, even when operating off of a Gaussian posterior.

Finally, a Molniya test case is simulated by Fig. 4. This test case shows that the filter manages to perform in non-circular orbit types with high eccentricity, and provides another example of comparable performance to the UKF when measurement densities are sufficient enough for it to generate a solution. This trial shows that the iPCE filter is capable of consistent and convergent results for a test case that the UKF is unable to handle.

Together, these trials provides evidence that the update and propagation systems investigated through this work show promise for orbital mechanics problems dealing with high levels of uncertainty or non-Gaussian probability distributions.

## CONCLUSIONS AND FUTURE WORK

All initial results from the iPCE filter and propagation tools agree with those generated through established techniques. The fusion of the intrusive Galerkin technique with the nonlinear polynomial-based filtering methods presented in 27 shows great potential towards the further development of an efficient and accurate filter which can account for nonlinear dynamical systems facing the need for precise uncertainty quantification.<sup>27</sup>

Current capabilities have shown the filter to be consistent and extensible to measurement sparse, nonlinear systems, but the development of the filter is far from complete. Alternative methods for coefficient truncation after the measurement update is performed, such as reducing the degree of the PCE back to the prediction degree without a Gaussian simplification, as well as investigation of updates that are even higher order could be considered for future projects. Additionally, applying the iPCF to other, more academic OD problems, such as the circular restricted three body problem, could help to evaluate the ability of the filter to improve ground-based orbit determination technologies.

## REFERENCES

- [1] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, Mar. 2004, pp. 401–422. Conference Name: Proceedings of the IEEE, 10.1109/JPROC.2003.823141.
- [2] K. J. DeMars, Y. Cheng, and M. K. Jah, "Collision Probability with Gaussian Mixture Orbit Uncertainty," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, p. 979–985, 10.2514/1.62308.
- [3] V. Vittaldev and R. P. Russell, "Space Object Collision Probability Using Multidirectional Gaussian Mixture Models," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 9, 2016, p. 2163–2169, 10.2514/1.g001610.
- [4] A. Mashiku, J. Garrison, and J. R. Carpenter, "Statistical Orbit Determination using the Particle Filter for incorporating Non-Gaussian Uncertainties.," *AIAA/AAS Astrodynamics Specialist Conference*, p. 5063.
- [5] X. Ning and J. Fang, "Spacecraft autonomous navigation using unscented particle filter-based celestial/Doppler information fusion," *Measurement Science and Technology*, Vol. 19, No. 9, 2008, p. 095203.
- [6] D. A. Reynolds, "Gaussian mixture models.," *Encyclopedia of biometrics*, Vol. 741, No. 659-663, 2009.
- [7] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control*, Vol. 17, Aug. 1972, pp. 439–448. Conference Name: IEEE Transactions on Automatic Control, 10.1109/TAC.1972.1100034.

- [8] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, Vol. 50, Feb. 2002, pp. 174–188. Conference Name: IEEE Transactions on Signal Processing, 10.1109/78.978374.
- [9] F. Xiong, S. Chen, and Y. Xiong, "Dynamic system uncertainty propagation using polynomial chaos," *Chinese Journal of Aeronautics*, Vol. 27, Oct. 2014, pp. 1156–1170, 10.1016/j.cja.2014.08.010.
- [10] B. Jia and M. Xin, "Active Sampling Based Polynomial-Chaos-Kriging Model for Orbital Uncertainty Propagation," *Journal of Guidance, Control, and Dynamics*, Vol. 44, May 2021, pp. 905–922. Publisher: American Institute of Aeronautics and Astronautics, 10.2514/1.G005130.
- [11] B. Jia and M. Xin, "Short-Arc Orbital Uncertainty Propagation with Arbitrary Polynomial Chaos and Admissible Region," *Journal of Guidance, Control, and Dynamics*, Vol. 43, Apr. 2020, pp. 715–728. Publisher: American Institute of Aeronautics and Astronautics, 10.2514/1.G004548.
- [12] B. A. Jones and A. Doostan, "Satellite collision probability estimation using polynomial chaos expansions," *Advances in Space Research*, Vol. 52, No. 11, 2013, p. 1860–1875, 10.1016/j.asr.2013.08.027.
- [13] B. A. Jones, N. Parrish, and A. Doostan, "Postmaneuver Collision Probability Estimation Using Sparse Polynomial Chaos Expansions," *Journal of guidance, control, and dynamics*, Vol. 38, No. 8, 2015, pp. 1425–1437.
- [14] B. A. Jones, A. Doostan, and G. H. Born, "Nonlinear Propagation of Orbit Uncertainty Using Non-Intrusive Polynomial Chaos," *Journal of guidance, control, and dynamics*, Vol. 36, No. 2, 2013, pp. 430–444.
- [15] Z. Yu, P. Cui, and M. Ni, "A polynomial chaos based square-root Kalman filter for Mars entry navigation," *Aerospace Science and Technology*, Vol. 51, Apr. 2016, pp. 192–202, 10.1016/j.ast.2016.02.009.
- [16] Y. Xu, L. Mili, and J. Zhao, "A Novel Polynomial-Chaos-Based Kalman Filter," *IEEE Signal Processing Letters*, Vol. 26, Nov. 2018, pp. 9–13, 10.1109/LSP.2018.2879453.
- [17] J. Son and Y. Du, "Comparison of intrusive and nonintrusive polynomial chaos expansion-based approaches for high dimensional parametric uncertainty quantification and propagation," *Computers & chemical engineering*, Vol. 134, 2020, pp. 106685–.
- [18] F. Xiong, S. Chen, and Y. Xiong, "Dynamic system uncertainty propagation using polynomial chaos," *Chinese Journal of Aeronautics*, Vol. 27, No. 5, 2014, pp. 1156–1170, <https://doi.org/10.1016/j.cja.2014.08.010>.
- [19] D. Xiu, *Numerical Methods for Stochastic Computations*. Princeton: Princeton University Press, 2010, doi:10.1515/9781400835348.
- [20] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM journal on scientific computing*, Vol. 24, No. 2, 2002, pp. 619–644.
- [21] C. Soize and R. Ghanem, "Physical Systems with Random Uncertainties: Chaos Representations with Arbitrary Probability Measure," *SIAM J. Scientific Computing*, Vol. 26, 01 2004, pp. 395–410, 10.1137/S1064827503424505.
- [22] J. D. Jakeman, M. S. Eldred, and K. Sargsyan, "Enhancing L1-minimization estimates of polynomial chaos expansions using basis selection," *Journal of Computational Physics*, Vol. 289, 2015, pp. 18–34.
- [23] O. M. Knio and O. P. L. Maître, "Uncertainty propagation in CFD using polynomial chaos decomposition," *Fluid Dynamics Research*, Vol. 38, sep 2006, p. 616, 10.1016/j.fluidyn.2005.12.003.
- [24] Z. Liu, D. Lesselier, B. Sudret, and J. Wiart, "Surrogate modeling based on resampled polynomial chaos expansions," *Reliability Engineering & System Safety*, Vol. 202, 2020, p. 107008.
- [25] O. P. Le Maître and O. P. Knio, *Scientific Computation: Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics*. Springer, 2010.
- [26] R. G. Ghanem and P. D. Spanos, "Spectral stochastic finite-element formulation for reliability analysis," *Journal of Engineering Mechanics*, Vol. 117, No. 10, 1991, pp. 2351–2372.
- [27] S. Servadio, R. Zanetti, and B. Jones, "Nonlinear Filtering With a Polynomial Series of Gaussian Random Variables," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. PP, 10 2020, pp. 1–1, 10.1109/TAES.2020.3028487.