AAS 24-307

NONLINEAR ESTIMATION WITH INTRUSIVE POLYNOMIAL CHAOS AND MULTI-FIDELITY METHODS

Zee R. Toler*and Brandon A. Jones†

Linear filtering methods cannot always provide consistent estimates for systems with nonlinear dynamics and measurement models. This has led to the application of nonlinear uncertainty quantification (UQ) and filtering to current orbit determination challenges. Polynomial Chaos Expansions (PCEs) are capable of modeling non-Gaussian uncertainty under sparse measurements and long propagation times. Multi-fidelity methods, where a low number of high-fidelity samples are used to correct a lower fidelity solution, have also seen use in UQ to balance efficiency and accuracy. This work improves upon the authors' nonlinear intrusive PCE-based filter with a multi-fidelity time update applied to the perturbed orbits problem.

INTRODUCTION

Orbit determination, navigation, and orbital rendezvous are problems with well defined solution spaces in near-Earth orbit for traditional applications, but techniques frequently applied in two-body dynamic systems may not perform as well (or may fail outright) in the more general n-body case. Many state of the art techniques are prone to failure when applied to three-body systems or to two-body orbits prone to rapid growth in uncertainty due to dynamic nonlinearities (exacerbated in eccentric orbits with close periapse passage, as in the case of the Molniya orbit).[1–3] Measurement sparsity is another challenge that leads to non-Gaussian probability distribution functions (PDFs) due to long propagation times and limited information. Common linear filtering techniques, even those like the Unscented Kalman Filter (UKF), which calculates the posterior mean and covariance using higher order information through the unscented transform,[4] struggle in cases where the ability to model non-Gaussian probability is required. While work has been underway, especially during the past few years, to apply current state of the art algorithms to these orbital regimes, reliable solutions for orbit determination outside of sampling or particle-based solutions are still needed.[3,5–8] As method complexity increases, however, so typically does computational demand. For applications like live orbit determination for mission operations, spacecraft navigation, or collision probability estimation, a solution able to model non-Gaussian probability with a low runtime may be desired. Polynomial chaos expansions (PCEs) and multi-fidelity methods, two techniques commonly employed by the uncertainty quantification (UQ) community, both have attractive characteristics when considering potential solutions to these challenges; PCEs may be used to characterize non-Gaussian uncertainty and model high-fidelity systems at a cost lower than what is typical of pure high-fidelity sampling methods, and multi-fidelity methods may be employed to further reduce the implementation challenges of this technique when considering more dynamically complex systems. This work

*Graduate Research Assistant, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, 2617 Wichita St, Austin, TX 78712

†Associate Professor, Aerospace Engineering and Engineering Mechanics, UT Austin, 2617 Wichita St, Austin, TX 78712

presents an approach to nonlinear filtering that leverages PCEs with multi-fidelity propagation to achieve a filter robust to perturbed dynamic systems, long propagation time between measurements, and accurate estimation of non-Gaussian PDFs.

Polynomial chaos expansions are situated at a convenient intersection of accuracy and efficiency, given that they are less expensive than Monte Carlo and other sampling methods and their nature as a surrogate method — a type of technique that provides a an analytical representation of a solution rather than of a series of points.[9] PCEs also provide analytic solutions to the moments of the PDF of our quantities of interest.[10–12] PCEs are a well established technique in the UQ community, and have more recently gathered attention for astrodynamics applications. Jones et al. used non-intrusive Polynomial Chaos Expansions (niPCEs) for applications such as conjunction assessment, and demonstrated a marked improvement in computational efficiency with an acceptable loss of accuracy compared to Monte-Carlo sampling.[13–15] These efforts considered niPCEs, yet intrusive methods could hold advantages in runtime and extensibility over non-intrusive chaos.[16,17] Note that several filtering methods that use PCEs exist in the literature due to their appealing qualities as a surrogate; however, these models either rely on non-intrusive chaos to accomplish the prediction step, or do not fully utilize a nonlinear, higher order measurement update.[18–20] Additional efforts in PCE-based uncertainty quantification and filtering endeavors continue to reach the field of orbital mechanics, with PC Kringing garnering attention for showing performance better than or roughly equal to that of each method considered individually.[21,22] While this method shows clear success at approximating the non-Gaussianity of orbit determination solutions, it does not provide an analytic solution to the probability moments or the Sobol indices, which are two advantages to the filtering approach this paper describes.

Multi-fidelity methods are a common technique in UQ to reduce runtime while retaining simulation accuracy.[23,24] Multi-fidelity methods are a class of techniques that typically rely on an expensive, high fidelity model in combination with a cheaper, low fidelity model. By applying corrections to the lower fidelity solution based on knowledge obtained from a smaller number of high fidelity computations, the accuracy of a solution can be bolstered without sacrificing all the runtime efficiency of the lower fidelity model. These methods have continued to gain attention in the astrodynamics community over the last five years as a way to preserve knowledge of highly perturbed, dynamically complex systems to achieve accurate orbit determination and navigation solutions at lower cost.[25,26] In the broader UQ community, multi-fidelity work has been published on niPCE ensemble Kalman filtering algorithms in the past[27,28] as well as on the combination of Taylor polynomials and differential algebra with multi-fidelity methods.[29,30]

The PCE filtering algorithm presented here was initially demonstrated in reference 31,[31] where the Intrusive Polynomial Chaos Filter (IPCF) was tested in a Sun-synchronous orbit and showed convergent behavior in cases where the UKF was unable to produce a consistent solution. The primary technical additions provided by this paper are a multi-fidelity augmentation to the filter's propagation step and the consideration of perturbed dynamics models.

This paper provides the necessary background information to understand intrusive PCEs, the construction of a polynomial-based quadratic estimator, and the relevant introductory information to understand the use of multi-fidelity propagation in the filter. By including a small number of point-propagated high fidelity samples, the propagated prior PCE may be corrected for perturbations present in the dynamic system that are challenging to include in the intrusive polynomial chaos propagator. Current results show that multi-fidelity dynamics are able to successfully compensate for drag, solar radiation, third body, and gravity gradient perturbations in a two-body system. Two tests cases are considered to provide these results, both utilizing range measurements from three simulated Geosynchronous spacecraft. The first test case considers a low-Earth orbit (LEO) modified from the original Sun-synchronous case that includes a J_2 gravity gradient perturbation only; the latter considers a Molniya orbit with gravity gradient, drag, third body, and solar radiation pressure perturbations to demonstrate the multi-fidelity augmentation of the IPCF.

PROBLEM DEFINITION

In this paper, we seek to solve the orbit determination problem in a way that is robust to non-Gaussian uncertainty. Typically, this requires the modeling of two important processes; the propagation of the state and the modeling of some measurement data. Here, we define the propagation problem as

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1}\right), \tag{1}$$

where $\mathbf{x}_k$ is $\mathbf{x}(t)$ at some time $t = t_k$, $\mathbf{x}_{k-1}$ represents the previous state, and $\boldsymbol{\theta}_{k-1}$ is some process noise sample assumed to be additive to the accelerative state. Finally, the dynamics model propagating the previous state to the current prediction is denoted as $\mathbf{f}\left(\mathbf{x}_{k-1}\right)$. A process to compute a predicted measurement to model data being received is also necessary. We define

$$\mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \boldsymbol{\epsilon}_k, \tag{2}$$

where $\mathbf{z}_k$ represents the computed measurement at $t = k$, $\boldsymbol{\epsilon}_k$ represents some additive measurement noise, and $\mathbf{h}\left(\mathbf{x}_k\right)$ is a model that computes a predicted measurement from the current state prediction. More traditional attempts at solving the dynamic estimation problem consider $\mathbf{x}\left(t\right)$ as a random variable and subsequently derive estimators through minimizing or maximizing specific performance metrics, such as the minimum mean square error. Here, we seek to solve the estimation problem representing the estimate in the form of a random variable, $\hat{\mathbf{X}}\left(t, \boldsymbol{\xi}\right)$, that is a function of both time and some random inputs, $\boldsymbol{\xi}$.

To define these random inputs, we take $(\Omega, \mathcal{F}, \mathcal{P})$ as a probability space, where Ω represents the sample space and $\mathcal{P}$ is the probability, or measure, on the σ-field $\mathcal{F}$ satisfying the three axioms of probability over Ω. Then, we define $\boldsymbol{\xi}$ as a random variable on $(\Omega, \mathcal{F}, \mathcal{P})$ where $\boldsymbol{\xi} : \Omega \rightarrow \Gamma^d \subseteq \mathbb{R}^d$. In other words, $\boldsymbol{\xi}$ is a mapping from the sample space to the dimension d state space of independent, identically distributed random variables representing the input uncertainties to the problem. For our case, we consider all random inputs to be Gaussian distributed with zero mean and unit variance, or $\boldsymbol{\xi} \sim \mathcal{N}(0,\ I_{dxd})$.

The remainder of this paper seeks to present a filtering algorithm considering a PCE surrogate, $\hat{\boldsymbol{X}}(t, \boldsymbol{\xi})$, as the estimate, rather than performing the derivation of an estimator via a simple random variable only depending on time, $\hat{\mathbf{x}}\left(t\right)$. The arithmetic details necessary for PCE manipulation are described at length, as is the construction of a polynomial based filter using the surrogate to describe the solution estimate. While describing Eqs. 1 and 2 using a PCE demonstrates marked advantages, it also necessitates a more difficult implementation due to these arithmetic considerations, which is why multi-fidelity augmentation lends itself to inclusion in the IPCF.

POLYNOMIAL CHAOS

A PCE is an expansion of an orthogonal polynomial basis as a function of the random inputs of the stochastic system; the coefficients of the PCE are the projection of the modeled function onto the basis. The PCE solution is a sum of the products of coefficients and evaluated polynomials, represented via

$$\hat{\boldsymbol{X}}(t, \xi) = \sum_{\mathbf{k}=0}^{P} \boldsymbol{c}_{\mathbf{k}}(t) \psi_{\mathbf{k}}(\boldsymbol{\xi}), \tag{3}$$

where the variables $\hat{\boldsymbol{X}}(t, \boldsymbol{\xi})$, $\boldsymbol{c}_{\boldsymbol{k}}(t)$, and $\psi_{\boldsymbol{k}}(\boldsymbol{\xi})$ denote the estimated solution given by the PCE, the $\mathbf{k}^{\text{th}}$ coefficient of the PCE, and the $\mathbf{k}^{\text{th}}$ basis function evaluated at a random input $\boldsymbol{\xi}$. If we assume some true solution $\boldsymbol{X}(t)$ to be a function of the time of interest, $t \in \left[0, t_f\right]$, then the PCE is a surrogate, denoted as $\hat{\boldsymbol{X}}(t, \boldsymbol{\xi})$, that represents the solution's projection onto the basis of orthonormal polynomials as functions of $\boldsymbol{\xi}$. The degree, p, combined with the stochastic dimension, d, dictate the number of PCE terms,

$$P := \frac{(p+d)!}{p!d!}. \tag{4}$$

Note that these methods are subject to the curse of dimensionally, as when d and p increase, the corresponding increase in P is nonlinear. A higher system degree implies the generation of a more capable surrogate, usually resulting in a better approximation of the PDF of the quantities of interest (QoI).

For the PCE basis functions in this work, the authors have selected Hermite polynomials due to their relationship to the PDF of the Gaussian distribution. Recall that classically orthogonal polynomials are defined using the inner product with respect to a weighting function,

$$\langle \psi_m(\xi), \psi_n(\xi) \rangle = \int_{-\infty}^{\infty} \psi_m(\xi) \psi_n(\xi) w(\xi) d\xi = 0 \quad \forall m \neq n, \quad w(\xi) = e^{-\frac{\xi^2}{2}}, \tag{5}$$

where $w(\xi)$ is the scaled Gaussian probability distribution function for Hermite polynomials. This relationship is advantageous relative to Taylor expansions in that PCEs have no radius of convergence and may accurately describe distribution tails without truncation (subject to the accuracy of the expansion coefficients). Although they are not considered here, it should be noted that other random inputs, and thus polynomial types, may also be used in the construction of a PCE.[12]

The indexing variable $\mathbf{k}$ in Eq. 3 represents the expansion as a sum over one index. While this is a convenient notation for many computational operations, each element in the expansion is better described as a *multi-index*. In this case, each index contains multiple values with each element corresponding to the orthogonal polynomials multiplied together to create that particular element of the basis. Just as written in Eq. 3, the maximum number of terms is still a function of p and d. Then each multi-index $\mathbf{k}$ can be unpacked to define an element of the basis as

$$\psi_{\mathbf{k}}(\boldsymbol{\xi}) = \phi_{n_1}(\xi_1)\, \phi_{n_2}(\xi_2) \cdots \phi_{n_d}(\xi_d) \quad \forall\, \boldsymbol{n} \ni \sum_{i=1}^{d} n_i \leq p, \quad 0 \leq n_i \leq d, \tag{6}$$

and would be composed of d values of n_i, and a $\mathbf{k}$ would then exist for each possible product of the basis functions according to Eq. 6. These polynomials, $\phi(\xi)$, are univariate functions of the random inputs to the system and represent orthogonal polynomials of choice.

The PCE coefficients may be used to analytically calculate the moments of a PDF; using the notation in Eq. 3, $\mathbf{c_0}$ would correspond to the mean,

$$\mu_f(t) \triangleq \mathbb{E}\left[\hat{\boldsymbol{X}}(t, \boldsymbol{\xi})\right] = \mathbb{E}\left[\sum_{k=0}^{P} \boldsymbol{c}_{\boldsymbol{k}}(t) \psi_{\boldsymbol{k}}(\boldsymbol{\xi})\right] = \mathbf{c_0}(t), \tag{7}$$

and the covariance can be expressed algebraically as

$$\boldsymbol{\Sigma}_{\boldsymbol{XX}}(t) = \mathbb{E}\left[(\hat{\boldsymbol{X}}(t,\boldsymbol{\xi}) - \overline{\boldsymbol{X}}(t,\cdot))(\hat{\boldsymbol{X}}(t,\boldsymbol{\xi}) - \overline{\boldsymbol{X}}(t,\cdot))^T\right], \tag{8}$$

or simply as the sum of squares of the non-0$^{\text{th}}$ order terms of the expansion, where each term in the resultant covariance matrix is calculated as

$$\Sigma_{ij} = \sum_{k=1}^{P} c_{i,k} c_{j,k} \quad i,j \in [0,d], \tag{9}$$

where k is the index of each term in the expansion and i and j denote the indexing for the expansion's stochastic dimension. Note that here, k is a scalar representation of the aforementioned multi-index, $\mathbf{k}$, with terms having index values from 0 to P. For two different multivariate PCEs, $\hat{\boldsymbol{Y}}(t,\boldsymbol{\eta})$ and $\hat{\boldsymbol{X}}(t,\boldsymbol{\xi})$ having coefficients $\mathbf{b}$ and $\mathbf{c}$, each term of their cross covariance matrix would be equal to

$$\Sigma_{ij} = \sum_{k=1}^{P} c_{i,k} b_{j,k} \quad i \in \left[0, d_{\hat{\boldsymbol{X}}}\right] \quad j \in \left[0, d_{\hat{\boldsymbol{Y}}}\right], \tag{10}$$

where Σ_{ij} is an element of $\boldsymbol{\Sigma}_{\mathbf{XY}}$, a matrix of shape $d_{\hat{\boldsymbol{X}}}$ by $d_{\hat{\boldsymbol{Y}}}$. Terms c_i and b_j represent the i^{th} or j^{th} coefficient vector for the corresponding multivariate PCE. Higher order probability moments may also be calculated from the PCE coefficients if desired. The ability to analytically produce the moments of the PCE is one advantage of this work. Additionally, because the moments have an analytic relationship to the surrogate, a PCE can be created using a mean and covariance without the use of sampling techniques. This extends to other uncertainty quantification measures as well, such as the Sobol indices, which may also be computed as an analytic function of the PCE coefficients using knowledge of the PCE multi index.[32]

Random sampling is one approach to solve for the coefficients in cases where analytic solutions become burdensome, such as when a solution is non-Gaussian and would require more tedious calculation. By evaluating $\boldsymbol{\psi}$, some polynomial basis belonging to a PCE, $\boldsymbol{Y}(t,\boldsymbol{\xi})$, at instances of the random inputs, $\boldsymbol{\xi}$, the data matrix H may be computed. These same random inputs are then used to generate random samples from the surrogate, which have the relationship

$$\underbrace{\begin{bmatrix} \psi_0(\xi_1) & \cdots & \psi_P(\xi_1) \\ \psi_0(\xi_2) & \cdots & \psi_P(\xi_2) \\ \vdots & \ddots & \vdots \\ \psi_0(\xi_N) & \cdots & \psi_P(\xi_N) \end{bmatrix}}_{\boldsymbol{H}} \underbrace{\begin{bmatrix} \boldsymbol{c}_0^T(t_f) \\ \boldsymbol{c}_1^T(t_f) \\ \vdots \\ \boldsymbol{c}_P^T(t_f) \end{bmatrix}}_{\boldsymbol{C}} = \underbrace{\begin{bmatrix} \boldsymbol{Y}^T(t_f,\boldsymbol{\xi}_1) \\ \boldsymbol{Y}^T(t_f,\boldsymbol{\xi}_2) \\ \vdots \\ \boldsymbol{Y}^T(t_f,\boldsymbol{\xi}_N) \end{bmatrix}}_{\boldsymbol{Y}}, \tag{11}$$

or, in matrix notation,

$$\boldsymbol{HC} = \boldsymbol{Y}, \tag{12}$$

which may be solved to produce estimated coefficients, $\hat{\boldsymbol{C}}$, for $\boldsymbol{Y}(t,\boldsymbol{\xi})$ using the normal solution to the least squares problem,

$$\hat{\boldsymbol{C}} = \left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Y}. \tag{13}$$

A least squares approach with random sampling makes the most sense in the context of non-intrusive polynomial chaos, or when refitting one PCE to a PCE having a different basis. When utilizing a sampling approach, non-intrusive work often attempts to leverage coefficient sparsity or specific techniques to ensure fast evaluation.[33–35] These techniques are described as non-intrusive because they leave the dynamics or particulars of the stochastic system as a black box, meaning no alterations to the equations of motion or model of interest are required to produce a surrogate. Rather, random samples drawn from the input distribution, $\boldsymbol{\xi}$, and are scaled and evaluated using the model of interest to produce realizations of $\hat{\boldsymbol{Y}}(t, \xi)$. After evaluating the data matrix H with these random input samples, a least squares estimate of the coefficients may be computed using the surrogate realizations according to Eq. 13.

The Galerkin Method

Rather than use more common non-intrusive methods, we use intrusive chaos to develop the filtering algorithm presented here. This is because intrusive chaos provides an analytic solution to the PCE coefficient derivatives, which may then be integrated over time without reliance on random samples and point propagation. This work uses the Galerkin method, a spectral projection technique able to prevent secular growth in the terms of the PCE basis after continued algebraic operations, to derive differential equations for the PCE coefficients. This means that the surrogate may be propagated forwards and backwards through time analytically without sampling error changing the filtering result; for two runs using the same data, the exact same result would be produced.[11,36] The differential equations

$$\dot{c}_i = \varphi\left(i, \mathbf{c}\left(t\right)\right), \qquad \forall i \in [0, P], \tag{14}$$

describe the rate of change of the coefficients, $\mathbf{c}$. While the ability to directly propagate the coefficients is useful, it does come with disadvantages; most notably, for each given test case, φ must be derived for each dynamical system (along with some $\mathbf{h}\left(i, \mathbf{c}\left(t\right)\right) \forall i \in [0, P]$ for each measurement model) of interest. To formulate these equations, the Galerkin projection is utilized to reconstruct the algebraic operations composing the equations of motion in a way that is suitable for PCEs. While this method generalizes to PCEs that have different bases from one another, for the sake of clarity this paper assumes the projection is performed between two identical bases onto a third, also identical basis. To begin discussion, we take a simple product operation between two random variables, u and v, such that

$$s = uv. \tag{15}$$

The function of this projection method is encapsulated well by investigation of this simple nonlinear operation utilizing PCEs. So, to define the Galerkin product, we begin with two different PCEs,

$$u(\xi) = \sum_{k=0}^{P} u_k \psi_k(\xi), \quad \text{and} \quad v(\xi) = \sum_{k=0}^{P} v_k \psi_k(\xi). \tag{16}$$

Here, we want to multiply the expansions $u\left(\xi\right)$ and $v\left(\xi\right)$. Attempting this multiplication naively results in a double sum over the product of each element,

$$s(\xi) = \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \psi_i(\xi) \psi_j(\xi). \tag{17}$$

which clearly contains more terms (and terms of a higher order) than those of the bases of both expansions before this operation has been performed. Instead, the Galerkin projection considers the projection of this product, or the PCE $s(\xi)$, onto the basis $\boldsymbol{\psi}$. This process results in a product that can be abbreviated as

$$s_k = \frac{\langle s, \psi_k \rangle}{\langle \psi_k^2 \rangle} = \sum_{i=0}^{\mathrm{P}} \sum_{j=0}^{\mathrm{P}} u_i v_j C_{ijk} \quad \forall i \in [0, P]\,, \tag{18}$$

with s_k denoting the k^{th} coefficient of P for the resultant PCE $s(\xi)$ and C_{ijk} denoting the Galerkin product tensor, a sparse, three dimensional tensor computed from the normalized inner product of each basis element.[11] The tensor,

$$C_{ijk} \equiv \frac{\langle \psi_i \psi_j \psi_k \rangle}{\langle \psi_k \psi_k \rangle}, \tag{19}$$

can then be computed for a known d, p, and orthogonal polynomial selection as a one time cost. In this same fashion, nonlinear operations (inversion, division, and any polynomial formulation) may be computed pseudo-spectrally without secular growth in the number of terms after each operation.[11]

Extending Galerkin Operations

The Galerkin method provides the fundamental tools to construct intrusive equations of motion, measurement models, and other operations, but is not able to replicate non-polynomial functions without the use of numeric solvers. Even in the case of the square root, numeric intervention is necessary to produce a result unless the Galerkin method is extended using alternative strategies, such as those presented here. For the square root operation

$$\sqrt{u}(\xi)\sqrt{u}(\xi) = u(\xi), \tag{20}$$

a numerical solution that minimizes the residual of the matrix equation

$$\begin{pmatrix} \sum_{j=0}^{\mathrm{P}} C_{j00}\sqrt{u}_j & \dots & \sum_{j=0}^{\mathrm{P}} C_{j\mathrm{P}0}\sqrt{u}_j \\ \vdots & \ddots & \vdots \\ \sum_{j=0}^{\mathrm{P}} C_{j0\mathrm{P}}\sqrt{u}_j & \dots & \sum_{j=0}^{\mathrm{P}} C_{j\mathrm{PP}}\sqrt{u}_j \end{pmatrix} \begin{pmatrix} \sqrt{u}_0 \\ \vdots \\ \sqrt{u}_{\mathrm{P}} \end{pmatrix} = \begin{pmatrix} u_0 \\ \vdots \\ u_{\mathrm{P}} \end{pmatrix} \tag{21}$$

is necessary to solve for the square root coefficients.[11] Computation time, difficulty of implementation, and lack of function coverage are all notable drawbacks to the Galerkin method. Attempts to mitigate these drawbacks have led to promising work utilizing finite integration of the function derivative or a Taylor series-based approach to model functions of interest that the Galerkin method would not normally support.[37] The former is more reliable due to a lack of radius of convergence seen in the PCE Taylor series, but requires greater effort to implement. Due to the frequent use of non-polynomial functions in the astrodynamics community (largely due to the prevalence of angle measurements, such as right ascension and declination), implementing efficient approximations of these functions is paramount. For the work presented in this paper, the Taylor series approach has been implemented in the authors' software, extending the functionality of the IPCF to include trigonometric functions, their inverses, the square root, and other non-polynomial functions. Recall that for a function $\tau(x)$ infinitely differentiable at a, the Taylor series describing the solution is

$$\tau(x) = \sum_{n=0}^{\infty} \frac{\tau^{(n)}(a)}{n!}(x-a)^n, \tag{22}$$

which can be truncated according to the user's tolerance requirements. To apply this formulation to some PCE, $u(\xi)$, undergoing some non-polynomial operation τ, a Taylor series is expanded about the mean of the input expansion, u_0, and the term exponentiated by n, $(x-a)$, can be described as the PCE with the mean element subtracted out, or

$$d(\xi) = u(\xi) - u_0 = \sum_{i=1}^{P} u_k \psi_k, \tag{23}$$

where higher powers are computed according to the Galerkin product, resulting in the Taylor series,

$$s(\xi) = \sum_{n=0}^{\infty} \frac{\tau^{(n)}(u_0)}{n!} d^n(\xi), \tag{24}$$

having radius of convergence of $|\widehat{a} - a_0| < |a_0|$.[37] When the series is difficult to express computationally, the derivatives of the function τ may be computed symbolically and stored as functions for later use. In Debusschere et al.'s work,[37] the error in an expansion having n terms is defined as the mean of the input expansion divided by the maximum term in the n^{th} series term. A tolerance value of 10^{-15} showed good performance in that work, so that same tolerance value was used by the authors here. Taylor series using PCEs are employed only for purposes of efficiency in this paper to reduce runtime costs associated with numerically solving the intrusive square root. Care should be taken only to employ these methods when input distributions are close to Gaussian or the surrogate degree is high enough to accurately model the input and output distributions.

THE INTRUSIVE POLYNOMIAL CHAOS FILTER

Leveraging the Galerkin method, the authors have developed a filtering algorithm able to estimate the state of a QoI using a nonlinear polynomial update.[31] The IPCF was conceived by leveraging work by Servadio et al. that demonstrated a quadratic update using Taylor polynomials.[38] These methods are readily extensible to orthogonal polynomial bases. By first using the Galerkin projection to render a set of differential equations for the PCE coefficients for a given system, a propagated a priori PCE may be obtained. The filtering algorithm may then render non-Gaussian PDFs by utilizing a nonlinear (quadratic) measurement update just as in Servadio et al.'s work with Taylor polynomials.[38] This section and the following sub-sections provide details on these components and their combination to produce the IPCF.

In the case of Taylor polynomials, Isserlis' theorem must be used to calculate the moments of the polynomial estimate and each expansion naturally has a radius of convergence and will fail to accurately model those PDFs with infinite tails (as seen even in a simple Gaussian distribution). PCEs represent an attractive opportunity to continue this work because of their direct relationship to user-defined random inputs and their coefficients' analytic relationship to the moments, as illustrated in Eqs. 7 and 8.

Mathematical detailings of the filter were previously presented in the authors' previous work,[31] and the derivation of a nonlinear polynomial update is discussed at length in Servario et al.'s work,[38] but they are reiterated here for clarity. This work's primary novel addition to the filter is a multi-fidelity augmentation to the time update, explored in detail below.

Figure 1 contains a high level overview of the software written by the authors to perform the filtering algorithm outlined by this section. The majority of the filter initialization and filtering architecture

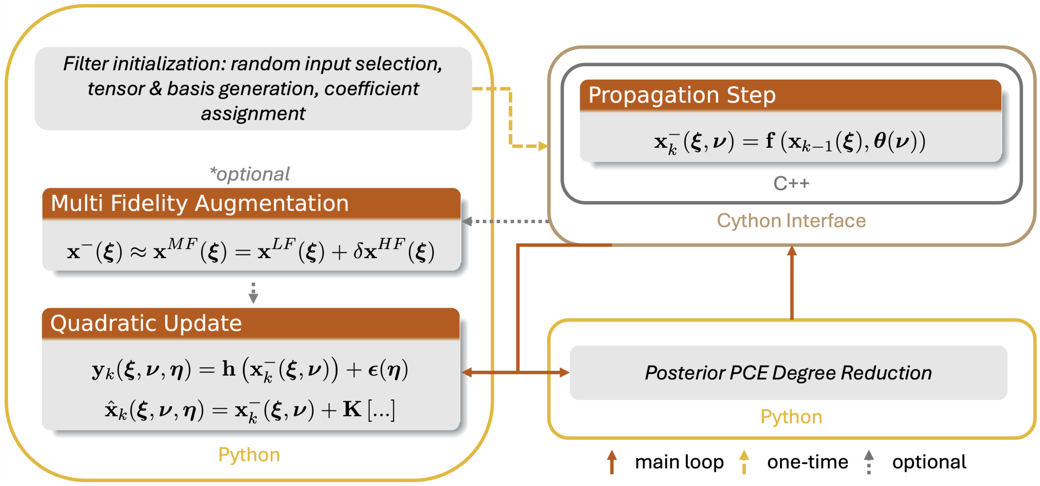


Figure 1. IPCF: software implementation

runs in Python 3, with limited dependence on Chaospy, a python package for polynomial chaos, for generation of the Galerkin tensor (currently, this tool is only used to symbolically compute the inner products of Hermite polynomials required for tensor generation).[39] The intrusive propagation is performed in C++ with a Cython interface providing scripting utility directly from the Python software.

The filtering algorithm runs in a loop similar to other estimation algorithms, with an intrusive polynomial chaos propagation step, an optional multi-fidelity addition to this step, a quadratic update step, creating a posterior PCE $2p_{pred}$ (where p_{pred} is the degree of the prediction PCE) in degree, and finally a polynomial truncation step, mean to reduce the posterior PCE back to p_{pred}. After reduction, the PCE is intrusively propagated until the next measurement time and the process repeats. The filtering process outlined below describes the manipulation of two PCEs, with $\boldsymbol{x}\,(\ldots)$ denoting the state PCE and $\boldsymbol{y}\,(\ldots)$ denoting the measurement PCE. Note that here the ellipses stand in for different combinations of random inputs depending on what filter operations have been completed (for example, the addition of a PCE depending on the process noise random inputs to a PCE depending on the random inputs to the prior). The measurement noise, $\boldsymbol{\epsilon}\,(\boldsymbol{\eta})$, is assumed to be additive, and the process noise PCE, $\boldsymbol{\theta}\,(\boldsymbol{\nu})$, is assumed additive to the accelerative state. Both $\boldsymbol{\epsilon}\,(\boldsymbol{\eta})$ and $\boldsymbol{\theta}\,(\boldsymbol{\nu})$, are assumed static, degree one ($d = 1$) PCEs, meaning they have a mean and variance only. Both noise PCEs are assumed to have a zero mean. A $+$ denotes a posterior estimate, $-$ denotes an a priori estimate, and all PCEs are assumed a function of time, t, with k denoting the current time step.

Filter Initialization

Filter initialization includes allocating random inputs for the problem, referred to here as $\boldsymbol{\xi}$, $\boldsymbol{\nu}$, and $\boldsymbol{\eta}$ for the state PDF, process noise PDF, and measurement error PDF, respectively. As described previously, these random inputs are assumed to be uncorrelated and Gaussian distributed with zero mean and unit variance, hence the PCE basis is comprised of Hermite polynomial products. The Galerkin tensor must be generated only one time for the same p and d values, meaning that for the same PCE degree, the circular restricted problem and the two body problem would utilize the same tensor. Tensor calculation is completed using Chaospy according to Eq. 18. Using the analytic relationship between the surrogate and the known PDF moments of the prior distribution, initial coefficients are also calculated at this initialization stage.

Propagation Step

Utilizing the Galerkin method described by earlier sections, the propagation step of the filter utilizes intrusive chaos to propagate an a priori PCE's coefficients to the time of interest. After intrusive propagation, the model determining the initial a priori surrogate describing the state, $\boldsymbol{x}_{k-1}^{-}(\boldsymbol{\xi})$, may be written as,

$$\mathbf{x}_k^{-}(\boldsymbol{\xi}, \boldsymbol{\nu}) = \mathbf{f}\left(\mathbf{x}_{k-1}(\boldsymbol{\xi}), \boldsymbol{\theta}(\boldsymbol{\nu})\right), \tag{25}$$

a function of both the state random inputs, $\boldsymbol{\xi}$ and the process noise random inputs, $\boldsymbol{\nu}$, due to the PCE addition of $\boldsymbol{\theta}(\boldsymbol{\nu})$ to the accelerative state during integration.

Previous work described the performance of the propagation step fully, and characterized its accuracy through moment comparison between Monte-Carlo simulation and lower degree PCEs (which are more suitable for use in the IPCF algorithm). These results, available in the authors' previous publication, demonstrate the ability of the PCE surrogate to model non-Gaussian PDFs far more accurately than what would be seen using the unscented transform.[31]

Multi-Fidelity Augmentation

The most recent addition to the IPCF is an optional multi-fidelity correction performed after the propagation step. The addition of a multi-fidelity correction to the filter improves the extensibility of the filter without incurring high implementation costs; because the Galerkin projection necessitates a more complex approach when implementing equations of motion, the ability to correct a propagated surrogate using limited point sampling could improve filter functionality over a variety of dynamic perturbations, such as solar radiation pressure or drag. To perform the multi-fidelity correction, some true solution is assumed for the corrected surrogate,

$$\mathbf{x}(\boldsymbol{\xi}, \boldsymbol{\nu}) = \mathbf{x}^{LF}(\boldsymbol{\xi}, \boldsymbol{\nu}) + \delta\mathbf{x}^{HF}(\boldsymbol{\xi}), \tag{26}$$

where $\mathbf{x}(\boldsymbol{\xi}, \boldsymbol{\nu})$ represents the surrogate and $\mathbf{x}^{LF} = \mathbf{x}^{+}(\boldsymbol{\xi})$ is the propagated a priori surrogate described in Eq. 25. $\delta\mathbf{x}^{HF}$ represents an correction surrogate equal to the difference between these two values. The goal of this step is to approximate $\delta\mathbf{x}^{HF}$ and recover a surrogate as close to the truth as necessary.

Beginning with $\mathbf{x}_{k-1}^{+}(\boldsymbol{\xi})$, the posterior estimate from the previous time step, the filter calculates

$$\bar{\mathbf{x}} = \mathbb{E}\left[\mathbf{x}_{k-1}^{-}(\boldsymbol{\xi})\right], \quad \mathbf{P} = \operatorname{cov}\left(\mathbf{x}_{k-1}^{-}(\boldsymbol{\xi})\right), \quad \text{and} \quad \mathbf{P} = \mathbf{L}\mathbf{L}^{\top}, \tag{27}$$

Which analytically provides a mean and covariance for sampling. Next, the filter generates $N_{samples}$ realizations of the random input vector $\boldsymbol{\xi}$, referred to as $\boldsymbol{\xi}'$. These samples are drawn, scaled, and propagated, producing

$$\mathbf{x}^{HF}(\boldsymbol{\xi}') = \mathbf{f}_{HF}\left(\bar{\mathbf{x}} + \mathbf{L}\boldsymbol{\xi}'\right), \tag{28}$$

i.e. $N_{samples}$ high fidelity points via point propagation of the scaled samples. After propagation, the algorithm approximates the correction surrogate as a function of the samples $\boldsymbol{\xi}'$ to produce $\delta\mathbf{x}^{HF}(\boldsymbol{\xi}')$. To produce this surrogate, both surrogate realizations and evaluations of the polynomial basis are needed to perform a least squares fit via Eq. 13. First, samples from the low fidelity surrogate, $\mathbf{x}^{LF}(\boldsymbol{\xi}', \boldsymbol{\nu}')$, are drawn using the same samples, $\boldsymbol{\xi}'$, used to produce the high fidelity samples along with $\boldsymbol{\nu}'$, $N_{samples}$ process noise realizations. By subtracting the low fidelity PCE realizations from the propagated high fidelity samples, a set of errors in the realizations,

$$\delta\mathbf{x}^{HF}(\boldsymbol{\xi}', \boldsymbol{\nu}') = \mathbf{x}^{HF}(\boldsymbol{\xi}') - \mathbf{x}^{LF}(\boldsymbol{\xi}', \boldsymbol{\nu}'), \tag{29}$$

is produced. Finally, by evaluating the the polynomial basis of composition determined by d, the length of the vector $\boldsymbol{\xi}$, and the desired degree of correction PCE, p_{corr}, at the same random input realizations, the coefficients

$$\hat{C}_{\delta\mathbf{x}} = \left(H^T H\right)^{-1} H^T \delta\mathbf{x}^{HF}(\boldsymbol{\xi}', \boldsymbol{\nu}'), \tag{30}$$

of the correction surrogate, $\delta\mathbf{x}^{HF}(\boldsymbol{\xi}, \boldsymbol{\nu})$, are estimated. With $\delta\mathbf{x}^{HF}(\boldsymbol{\xi}, \boldsymbol{\nu})$ now known, the multi-fidelity surrogate is

$$\mathbf{x}(\boldsymbol{\xi}, \boldsymbol{\nu}) \approx \mathbf{x}^{MF}(\boldsymbol{\xi}, \boldsymbol{\nu}) = \mathbf{x}^{LF}(\boldsymbol{\xi}, \boldsymbol{\nu}) + \delta\mathbf{x}^{HF}(\boldsymbol{\xi}, \boldsymbol{\nu}), \tag{31}$$

and the filter proceeds to the quadratic measurement update with the augmented prior.

Currently, the surrogate addition only matches PCE terms that have a basis element present in both the correction surrogate and the propagated prior, meaning that for $p_{corr} > p_{pred}$, not every error term is incorporated into the solution. Despite this, higher degree correction surrogates can still impart more information than those of the prediction degree, since the added terms are calculated with additional modeling capability corresponding to their expansion degree.

Quadratic Update

The filter's quadratic update has been detailed in previous work by the authors and Servadio et al, but is explained here for clarity.[31,38] The quadratic update may be replaced with a polynomial update of higher order, but this has not been considered for implementation in the IPCF. Starting with a family of quadratic estimators having constraints $\mathbf{a}$ and gain matrices $\mathbf{K}_1$ and $\mathbf{K}_2$, a cost function

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbf{K}_1\mathbf{y} + \mathbf{K}_2\mathbf{y}^{[2]}, \tag{32}$$

may be defined. Adding and subtracting constants (mean values for the state, measurement, and measurement squared) results in a separate estimator family based on the measurement residual, $\mathbf{y} - \mathbb{E}\{\mathbf{y}\}$, the squared residual, $\mathbf{y}^{[2]} - \mathbb{E}\{\mathbf{y}^{[2]}\}$, and the mean of the state, $\mathbb{E}\{\mathbf{x}\}$,

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1^*(\mathbf{y} - \mathbb{E}\{\mathbf{y}\}) + \mathbf{K}_2^*\left(\mathbf{y}^{[2]} - \mathbb{E}\{\mathbf{y}^{[2]}\}\right), \tag{33}$$

for which there are two optimal gain values, $\mathbf{K}_1^*$ and $\mathbf{K}_2^*$, and an optimal value for $\mathbf{a}^*$. Utilizing the orthogonality principle, a linear system of equations may be produced to solve for each optimal coefficient. The optimal $\mathbf{a}^* = 0$, and

$$\begin{bmatrix} \mathbf{K}_1^* & \mathbf{K}_2^* \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{d\mathbf{x}d\mathbf{y}}^{\mathrm{T}} \\ \mathbf{P}_{d\mathbf{x}d\mathbf{y}^{[2]}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{P}_{d\mathbf{y}d\mathbf{y}} & \mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]}} \\ \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}} & \mathbf{P}_{d\mathbf{y}^{[2]}\mathbf{d}\mathbf{y}^{[2]}} \end{bmatrix}^{-1}, \tag{34}$$

results in the quadratic estimator

$$\hat{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1^* d\mathbf{y} + \mathbf{K}_2^* d\mathbf{y}^{[2]}, \tag{35}$$

for which a more complete derivation may be found in reference 38.[38] While the form of the estimator itself is the same, the implementation must be reworked according to the nature of the basis functions used. Here, the construction of a quadratic estimator using PCEs is provided in sequential order. First, the IPCF necessitates the calculation of a PCE for the measurement,

$$\mathbf{y}_k(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta}) = \mathbf{h}\left(\mathbf{x}_k^-(\boldsymbol{\xi}, \boldsymbol{\nu})\right) + \boldsymbol{\epsilon}(\boldsymbol{\eta}), \tag{36}$$

a functional of the a priori PCE, $\mathbf{x}_k^-(\boldsymbol{\xi}, \boldsymbol{\nu})$ and the measurement noise PCE $\boldsymbol{\epsilon}(\boldsymbol{\eta})$. With this, it is possible to compute a squared measurement PCE, $\mathbf{y}_k^2(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta})$. This is done using a type of Kronecker vector product,

$$\begin{gathered}\mathbf{y}_k^2(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{y}(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta}) \otimes \boldsymbol{y}(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta}) \triangleq y_k^2\psi_k = y_i\psi_i y_j\psi_j, \\ \text{where } i \in [0, P], \, j \in [i, P]\end{gathered} \tag{37}$$

with no repeating terms. Of particular note is that this process *squares the polynomial surrogate*, meaning a PCE of degree $2d_{pred}$ results from this step, where each individual product making up the vector product is still computed pseudo-spectrally according to the Galerkin projection. The software accommodates this by generating Galerkin tensor elements corresponding to the doubled (update) degree at filter initialization; after the measurement polynomial is squared, the expanded tensor is used in all Galerkin operations for the remainder of the filtering step. The squared measurement,

$$\tilde{\mathbf{y}}_k^{[2]} = \tilde{\mathbf{y}}_k \otimes \tilde{\mathbf{y}}_k, \tag{38}$$

is computed in the same way.

For the remainder of this section, $\bar{\mathbf{y}}_k$ denotes the mean of the PCE $\mathbf{y}_k(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta})$, while $\tilde{\mathbf{y}}_k$ denotes the k^{th} measurement currently being processed by the filter update. To complete the update, the filter must compute the covariance matrices comprising each gain value shown in Eq. 34. The mean values for the PCEs may be computed trivially according to Eq. 7, producing the propagated a priori mean $\bar{\mathbf{x}}_k^- = \mathbb{E}\left\{\mathbf{x}_k^-(\boldsymbol{\xi}, \boldsymbol{\nu})\right\}$, the measurement PCE mean $\bar{\mathbf{y}}_k = \mathbb{E}\{\mathbf{y}_k(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta})\}$, and the mean of the squared measurement PCE $\bar{\mathbf{y}}_k^{[2]} = \mathbb{E}\left\{\mathbf{y}_k^{[2]}(\boldsymbol{\xi}, \boldsymbol{\nu}, \boldsymbol{\eta})\right\}$. After calculating the expected values, the covariance and cross covariance matrices necessary for the gains may be computed from the PCE coefficients according to Eq. 8. Then,

$$\mathbf{P}_{\mathbf{yy}} = \mathbb{E}\left\{(\mathbf{y}_k - \bar{\mathbf{y}}_k)(\mathbf{y}_k - \bar{\mathbf{y}}_k)^{\mathrm{T}}\right\}, \tag{39}$$

the cross-covariance between the measurement and the squared measurement is

$$\mathbf{P}_{\mathbf{yy}^{[2]}} = \mathbb{E}\left\{(\mathbf{y}_{k+1} - \bar{\mathbf{y}}_k)\left(\mathbf{y}_k^{[2]} - \bar{\mathbf{y}}_k^{[2]}\right)^{\mathrm{T}}\right\}, \tag{40}$$

and the squared measurement covariance matrix is

$$\mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}^{[2]}} = \mathbb{E}\left\{\left(\mathbf{y}_k^{[2]} - \bar{\mathbf{y}}_k^{[2]}\right)\left(\mathbf{y}_k^{[2]} - \bar{\mathbf{y}}_k^{[2]}\right)^{\mathrm{T}}\right\}, \tag{41}$$

providing the three covariance measures necessary to construct the augmented measurement covariance. This can be expressed as the block matrix

$$\mathbf{P}_{\mathcal{Y}\mathcal{Y}} = \left[\begin{array}{c|c} \mathbf{P}_{\mathbf{yy}} & \mathbf{P}_{\mathbf{yy}^{[2]}} \\ \hline \mathbf{P}_{\mathbf{yy}^{[2]}} & \mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}^{[2]}} \end{array}\right], \tag{42}$$

and the cross-covariance of the measurement and squared measurement with the state is then

$$\mathbf{P}_{\mathbf{x}\mathcal{Y}} = \left[\mathbf{P}_{\mathbf{xy}} \quad \mathbf{P}_{\mathbf{xy}^{[2]}}\right], \tag{43}$$

which is computed in the same way as the augmented measurement covariance matrix as expressed in Eq. 10. Now that both gain matrices have been computed per Eq. 34, $\mathbf{K}_1^*$ and $\mathbf{K}_2^*$ are blocked

$$\mathbf{K} = \left[\begin{array}{cc} \mathbf{K}_1^* & \mathbf{K}_2^* \end{array}\right] = \mathbf{P}_{\mathbf{x}\mathcal{Y}}\mathbf{P}_{\mathcal{Y}\mathcal{Y}}^{-1}. \tag{44}$$

to compose an augmented gain matrix for use with the measurement residual and the squared measurement residual — the differences between the measurement and measurement polynomial and between the squared measurement and squared measurement polynomial. The full posterior PCE surrogate solution is then

$$\mathbf{x}_k^+(\boldsymbol{\xi},\boldsymbol{\nu},\boldsymbol{\eta}) = \mathbf{x}_k^-(\boldsymbol{\xi},\boldsymbol{\nu}) + \mathbf{K}\begin{bmatrix} \tilde{\mathbf{y}}_k - \mathbf{y}_k(\boldsymbol{\xi},\boldsymbol{\nu},\boldsymbol{\eta}) \\ \tilde{\mathbf{y}}_k^{[2]} - \mathbf{y}_k^{[2]}(\boldsymbol{\xi},\boldsymbol{\nu},\boldsymbol{\eta}) \end{bmatrix}, \tag{45}$$

the posterior mean is

$$\bar{\mathbf{x}}_k^+ = \mathbb{E}\left\{\mathbf{x}_k^+\right\} = \bar{\mathbf{x}}_k^- + \mathbf{K}\begin{bmatrix} \tilde{\mathbf{y}}_k - \bar{\mathbf{y}}_k \\ \tilde{\mathbf{y}}_k^{[2]} - \bar{\mathbf{y}}_k^{[2]} \end{bmatrix}, \tag{46}$$

and the posterior covariance is

$$\mathbf{P}_{\mathbf{xx},k} = \mathbb{E}\left\{\left(\mathbf{x}_k^+ - \bar{\mathbf{x}}_k^+\right)\left(\mathbf{x}_k^+ - \bar{\mathbf{x}}_k^+\right)^{\mathrm{T}}\right\}, \tag{47}$$

with both moments calculated according to equations 7 and 8.

Polynomial Reduction

The quadratic update produces a polynomial surrogate of degree $2d_{pred}$. Without altering the posterior PCE solution, it is clear that the degree of PCE used in the filter would quickly become both unsustainable for continued computation and overqualified to efficiently describe most stochastic systems. This is combated by reducing the degree of the polynomial surrogate to that of the original prediction degree. In more specific terms, the truncated posterior PCE becomes the prior PCE at time t_k for the next filtering step and will have the same degree and basis composition as that of the previous prior PCE.

Currently, the polynomial truncation step is completed by taking the posterior mean, Eq. 46, and the posterior covariance, Eq. 47, and computing PCE coefficients for a new, degree p_{pred} PCE. This is completed analytically using the relationship between these moments to the coefficients, similar to equations 7 and 8. This assignment is easy in the Gaussian case, since after performing the Cholesky decomposition, the multi-index may be used to identify coefficients pertaining to each random input and assign the elements of the decomposition by Cholesky matrix column to coefficient row with a multi-index value of 1 for the corresponding random input (where coefficients are stored in a P by d matrix).

Notably, this method only takes advantage of the Gaussian moments (mean and covariance) of the posterior in the formation of the new prior, meaning that only coefficients corresponding to a degree one PCE are filled with nonzero values. A more complex polynomial truncation method currently in progress would refit the posterior PCE coefficients to a polynomial of degree p_{pred} more completely through a least squares fit using Eq. 13. A naive fit can be performed by sampling the posterior PCE with some number of random input realizations, $\boldsymbol{\xi}_i \; \forall i \in N$, where N is a user-determined number of samples. Then, with the same random inputs, a new PCE basis of degree p_{pred} is evaluated, providing the data matrix $\mathbf{H}$. Eq. 13 is then solved to produce coefficients for the degree p_{pred} surrogate.

In addition to truncating higher-order information, this process is marginalizing the surrogate to remove dependence on the random inputs for the process and measurement noise, ν and η. Because of this marginalization, the PCE loses information conferred from the coefficients that describe the correlative relationship between the process or measurement noise and the prior state, causing the solution to become overly conservative. Ongoing work seeks to remedy this overly conservative estimate by matching the covariance of the posterior to the new a priori via a nonlinear programming solution.

RESULTS

Previous work used the unperturbed two body problem as the dynamics model for both filtering and propagation tests.[31] This work considers two body dynamics with J_2 and J_2 and J_3 perturbed equations of motion for similar test cases along with drag, solar radiation pressure, and third-body perturbations. The IPCF performance is contrasted with the performance of the UKF for a low-Earth orbit (LEO), modified from a Sun-synchronous case to be eccentric and lower altitude, and a Molniya orbit. A baseline test considering J_2 equations of motion demonstrates the performance of the two filters for a difficult orbit determination case with sparse measurements and a larger measurement noise variance to demonstrate the successful implementation of new dynamics models using the Galerkin method. Following this, another test case with similar characteristics and the same measurement model shows the performance of the multi-fidelity augmentation to the propagation step. This section presents Monte-Carlo consistency trials with associated RMSE statistics to explore filter behavior in each case.

$$\begin{array}{l}
\text{Sat } 1_{\text{ECEF}} = [\quad 1.0 \quad 0.0 \quad 0.0 \quad] \\
\text{Sat } 2_{\text{ECEF}} = [\quad \cos(120^\circ) \quad \sin(120^\circ) \quad 0.0 \quad] \\
\text{Sat } 3_{\text{ECEF}} = [\quad \cos(240^\circ) \quad \sin(120^\circ) \quad 0.0 \quad]
\end{array}$$

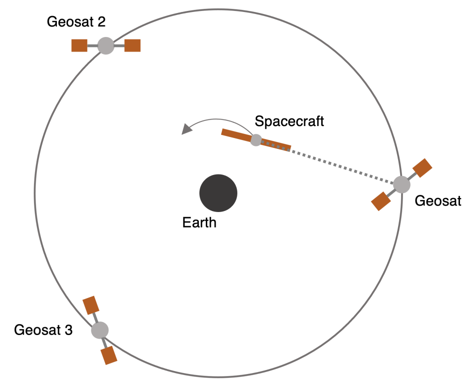


Figure 2. Geosatellite Simulation

This section considers results from a simulation of a spacecraft in a perturbed two-body orbit that receives range measurements relative to the location of three different Geosynchronous satellites, each set 120° degrees apart from one another. Whichever satellite is closest to the true state of the spacecraft at $t = t_k$ produces a range measurement for use in the filter. Range measurements provide a simple measurement framework while also ensuring a nonlinear measurement type, and the inclusion of three different measurement sources prevent problems with observability in the state. Each geosynchronous satellite state is propagated using the same dynamics model used for the true spacecraft state with no process noise (since the positions of the geosynchronous spacecraft are considered known).

Table 1 provides initial orbital elements for both test cases, a modified Sun-syncchronous orbit and a Molniya orbit. The gravitational parameter $\mu = 398600.4415$ km^3/s^2 , $R_{Earth} = 6378.1363$ km, $J_2 = 0.00108262668$, and $J_3 = -0.0000025327$. An exponential drag model is present in the

Table 1. Initial Keplarian orbital elements for both test cases

Case	a (km)	e	i (deg)	ω (deg)	Ω (deg)	ν (deg)
LEO	6845	0.2	97.7	0.0	0.0	0.0
Molniya	26553	0.737	63.4	0.0	270	0.0

Table 2. LEO Case Statistics

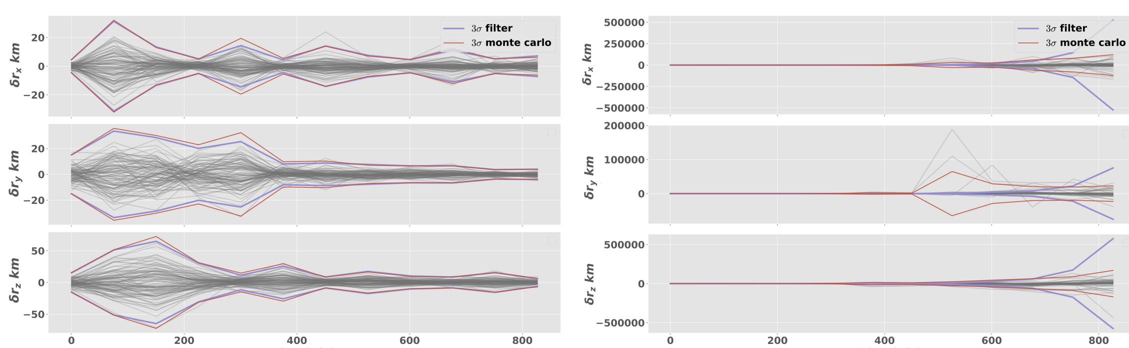
Value	Covariance Matrix
$\boldsymbol{P}_{xx}$	block[25 $\mathbf{I}_{3x3}$km^2, 2.5×10^{-4} $\mathbf{I}_{3x3}\frac{\text{km}^2}{\text{s}^2}$]
$\boldsymbol{Q}$	10^{-16} $\mathbf{I}_{3x3}\frac{\text{km}^2}{\text{s}^4}$
$\boldsymbol{R}$	2.5 km^2

Table 3. Molniya Case Statistics

Value	Covariance Matrix
$\boldsymbol{P}_{xx}$	block[15 $\mathbf{I}_{3x3}$km^2, 1.5×10^{-5} $\mathbf{I}_{3x3}\frac{\text{km}^2}{\text{s}^2}$]
$\boldsymbol{Q}$	10^{-16} $\mathbf{I}_{3x3}\frac{\text{km}^2}{\text{s}^4}$
$\boldsymbol{R}$	15 km^2

full-fidelity dynamics, with $C_r = 1.5$ and $A/m = 10.7\text{m}^2/\text{kg}$. Additional full-fidelity perturbations include an 8×8 spherical harmonics model and third body perturbations accounting for the Sun and Moon calculated using JPL DE430 ephemerides.[40,41]

First, this section establishes a performance baseline for each filter using the LEO test case. A similar test case was originally used on pervious work to showcase the performance of the filter with two body dynamics as significantly superior to that of the UKF for sparse measurements.[31] This first test, a comparison of the IPCF and UKF under two body dynamics, was repeated with a J_2 perturbation with results shown in Figure 3. These results demonstrate a divergent UKF solution compared to a PCE solution that still converges from the set of measurements provided to the filter. For both filter, one hundred trials were performed, where each trial varied the true solution and measurement errors, and both filters were provided the same measurement data as one another for each individual Monte Carlo trial. These results demonstrate that the IPCF remains viable with the addition of perturbations and serves to replicate earlier results with more complex dynamics. The RMSE of each filter in Table 4 show this difference in a numeric sense as well. Here, $p_{pred} = 2$ and $p_{update} = 4$. Measurements are provided to the filter every .8 periods, or every 75.1489 minutes with 12 measurements total (9.6 orbital periods). Statistics for this trial are provided in Table 2.



(a) IPCF (b) UKF

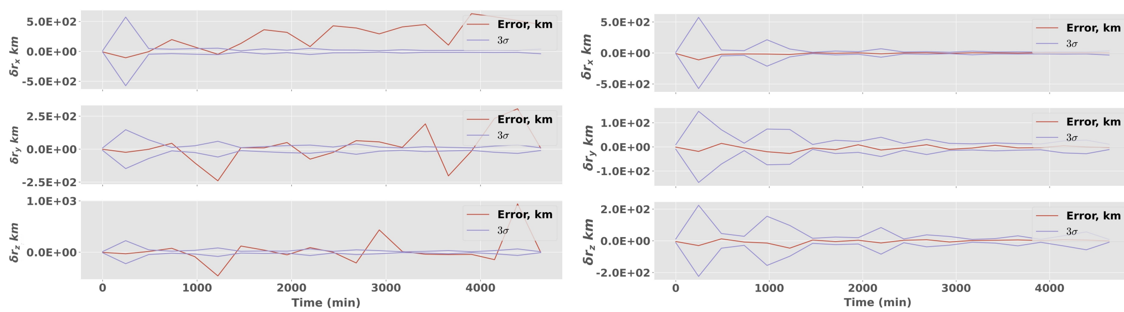
Figure 3. LEO test results: J_2 dynamics

Table 4. RMSE for the J_2 LEO Case

Value	UKF	IPCF
x km	9.741×10^3	4.069
y km	2.842×10^3	5.770
z km	1.214×10^4	9.041
$\dot{x}$ km/s	6.818	7.882×10^{-3}
$\dot{y}$ km/s	9.089×10^{-1}	6.658×10^{-3}
$\dot{z}$ km/s	5.927	7.144×10^{-3}

Table 5. RMSE for the multi-fidelity Molniya case

Value	UKF	IPCF
x km	4.597×10^1	4.048×10^1
y km	1.573×10^1	1.380×10^1
z km	2.414×10^1	1.932×10^1
$\dot{x}$ km/s	6.018×10^{-3}	5.183×10^{-3}
$\dot{y}$ km/s	2.985×10^{-3}	2.643×10^{-3}
$\dot{z}$ km/s	5.131×10^{-3}	4.357×10^{-3}



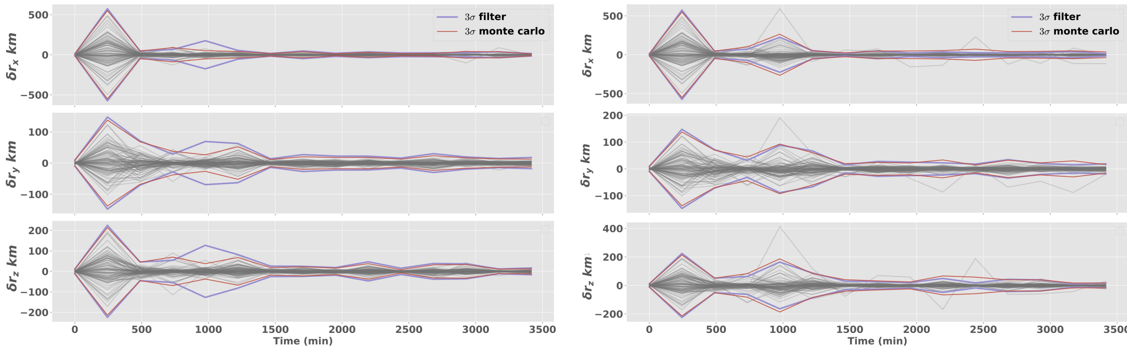
(a) J_2 and J_3 dynamics

(b) Multi-fidelity dynamics

Figure 4. Single filter trial with and without a multi-fidelity correction

The next trial considers the case of an orbit using the full fidelity dynamics described earlier (includes drag, spherical harmonics, etc). This particular test consists of conditions allowing for a UKF solution that begins to struggle to converge, but does still remain statistically significant. Measurements were received by the filtering algorithms every 0.67 orbits, or every 480.845 minutes with 15 measurements total (for 10 orbital periods). The PCEs utilized in the IPCF had degrees $p_{pred} = 2$, $p_{corr} = 3$, and $p_{update} = 4$. The impact of the multi-fidelity correction to the IPCF is visualized by a single corrected trial shown in Figure 4. Here, a solution propagated with the J_2 and J_3 perturbations is unable to provide an accurate state estimate that stays within the 3σ filter boundaries. The solution with the multi-fidelity correction shows a bounded state error. Table 3 contains the statistics that produced measurement samples and truth data for this run.

With the same measurement frequency and the same initial conditions, another 100 trial Monte Carlo consistency test was run to determine IPCF performance through a consistency trial and in an RMSE sense with multi-fidelity augmentation. Figure 5 shows the performance of each filter. The UKF shows a generally convergent solution, but does show edge cases out of the samples that deviate significantly from the 3σ uncertainty boundaries — the filter estimate is also shown to be overconfident, meaning it is likely approaching a scenario that would cause failure in the propagation step as uncertainty or measurement sparsity increased. The IPCF shows a converged, mostly bounded estimate — the figure shows a few outliers that briefly escape the 3σ boundaries of the filter, and, at a few points, the estimate is slightly overconfident (at the end of the trial in the x direction), but significantly less so than the UKF. This could be due to the random sampling used for the high fidelity point propagation. The most notable characteristic of the IPCF's performance is



(a) IPCF (b) UKF

Figure 5. Molniya orbit test results with multi-fidelity dynamics

the overly conservative 3σ boundary at around 1000 minutes time; this is likely due to a difference in the propagation strategies between the two methods, because of randomness in the multi-fidelity sampling selection, or due to the degree of PCE surrogate used to compose the correction PCE; future work will investigate ways to increase the impact of the error polynomial on the propagated prior's covariance. Table 5 shows the RMSE for each filter for the Monte Carlo trial. The RMSE is significantly less for the IPCF than the UKF indicating that the IPCF is also more accurate in the mean.

CONCLUSION

The authors' previous efforts have shown intrusive polynomial chaos to be a promising strategy for nonlinear, polynomial-based filtering.[31] Earlier work centered on two-body dynamics only, without considering the effects of dynamical perturbations; in this work, J_2 and J_3 equations of motion were developed using the Galerkin method and used in the propagation step of the IPCF. This development was combined with the addition of a multi-fidelity augmentation to the propagation step of the filter, allowing the IPCF to show convergent behavior and superior estimation to the UKF in an RMSE sense for an orbit under higher fidelity dynamics. More minor adjustments include the use of Taylor expansion using PCEs to approximate non-polynomial functions; currently, these advancements have only been used to mitigate the runtime cost of a numerical solution for the Galerkin square root in the quadratic update.

Future work will consider current advancements applied to additional test cases, measurement types, and dynamical systems. Chaotic systems like the Lorenz system as well as topical cases like popular orbital families in the circular restricted problem are both good candidates for use with the filtering algorithm. Completing angle measurement test cases through the use of already implemented Taylor expansion tools will also significantly improve the filter's extensibility. Further investigation into the multi-fidelity update and fine tuning the approach is also necessary, and could be done through more representative sample selection techniques and by modifying how the error and propagated prior PCEs are combined. Finally, additional methods of PCE reduction from the update degree to the prediction degree will be implemented and their performance characterized.

ACKNOWLEDGEMENTS

This work was supported by the NASA Space Technology Graduate Research Opportunities Fellowship, grant number 80NSSC23K1202.

REFERENCES

[1] J. L. Junkins, M. R. Akella, and K. T. Alfrined, "Non-gaussian error propagation in orbital mechanics," *Guidance and control 1996*, pp. 283–298, 1996.

[2] K. J. DeMars, R. H. Bishop, and M. K. Jah, "Entropy-Based Approach for Uncertainty Propagation of Nonlinear Dynamical Systems," *Journal of Guidance, Control, and Dynamics*, vol. 36, pp. 1047–1057, July 2013. Publisher: American Institute of Aeronautics and Astronautics.

[3] M. Thompson, A. Zara, C. Ott, M. Bolliger, E. Kayser, and D. Davis, "Comparisons of Filtering Algorithms for Orbit Determination in Near Rectilinear Halo Orbits," in *AAS/AIAA Astrodynamics Specialist Conference*, Aug. 2022.

[4] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, pp. 153–158, Ieee, 2000.

[5] B. Cheetham, "Cislunar Autonomous Positioning System Technology Operations and Navigation Experiment (CAPSTONE)," in *ASCEND 2021*, no. 2021-4128, American Institute of Aeronautics and Astronautics, Nov. 2021. https://arc.aiaa.org/doi/pdf/10.2514/6.2021-4128.

[6] C. Ott, M. Bolliger, M. Thompson, N. P. Ré, and D. C. Davis, "Range Biases, Measurement Noise, and Perilune Accuracy in Near Rectilinear Halo Orbit Navigation," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics, 2022.

[7] C. P. Newman, D. C. Davis, and S. Cryan, "A FOCUS ON NAVIGATION PERFORMANCE NEAR PERILUNE IN A NEAR RECTILINEAR HALO ORBIT FOR A CREWED STATION," in *AAS Guidance Navigation and Control Conference*, (Breckenridge, CO), p. 14, American Astronomical Society, 2022.

[8] M. Bolliger, M. Thompson, N. Ré, C. Ott, D. Davis, and N. Parrish, "Ground-Based Navigation Trades for Operations in Gateway's Near Rectilinear Halo Orbit," in *AAS/AIAA Space Flight Mechanics Conference*, Feb. 2021.

[9] N. Wiener, "The Homogeneous Chaos," *American Journal of Mathematics*, vol. 60, p. 897, Oct. 1938.

[10] R. G. Ghanem and P. D. Spanos, "Spectral stochastic finite-element formulation for reliability analysis," *Journal of Engineering Mechanics*, vol. 117, no. 10, pp. 2351–2372, 1991.

[11] O. P. Le Maître and O. P. Knio, *Scientific Computation: Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics*. Springer, 2010.

[12] C. Soize and R. Ghanem, "Physical systems with random uncertainties: Chaos representations with arbitrary probability measure," *SIAM J. Scientific Computing*, vol. 26, pp. 395–410, 01 2004.

[13] 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, pp. 430–444, 2013.

[14] B. A. Jones and A. Doostan, "Satellite collision probability estimation using polynomial chaos expansions," *Advances in Space Research*, vol. 52, no. 11, p. 1860–1875, 2013.

[15] 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, pp. 1425–1437, 2015.

[16] 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, pp. 106685–, 2020.

[17] F. Xiong, S. Chen, and Y. Xiong, "Dynamic system uncertainty propagation using polynomial chaos," *Chinese Journal of Aeronautics*, vol. 27, no. 5, pp. 1156–1170, 2014.

[18] 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, pp. 192–202, Apr. 2016.

[19] Y. Xu, L. Mili, and J. Zhao, "A Novel Polynomial-Chaos-Based Kalman Filter," *IEEE Signal Processing Letters*, vol. 26, pp. 9–13, Nov. 2018.

[20] E. D. Blanchard, A. Sandu, and C. Sandu, "A polynomial chaos-based kalman filter approach for parameter estimation of mechanical systems," in *ASME 2007 International Design Engineering Technical Conference*, (Las Vegas, Nevada), 2007.

[21] R. Schoebi, B. Sudret, and J. Wiart, "Polynomial-Chaos-based Kriging," Feb. 2015. arXiv:1502.03939 [stat].

[22] B. Jia and M. Xin, "Active Sampling Based Polynomial-Chaos–Kriging Model for Orbital Uncertainty Propagation," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 5, pp. 905–922, 2021. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G005130.

[23] B. Peherstorfer, K. Willcox, and M. Gunzburger, "Survey of multifidelity methods in uncertainty propagation, inference, and optimization," *SIAM Review*, vol. 60, no. 3, pp. 550–591, 2018.

[24] M. Giselle Fernández-Godino, "Review of multi-fidelity models," *Advances in Computational Science and Engineering*, vol. 1, no. 4, p. 351–400, 2023.

[25] B. A. Jones and R. Weisman, "Multi-fidelity orbit uncertainty propagation," *Acta Astronautica*, vol. 155, pp. 406–417, Feb. 2019.

[26] E. M. Zucchelli, E. D. Delande, B. A. Jones, and M. K. Jah, "Multi-Fidelity Orbit Determination with Systematic Errors," *The Journal of the Astronautical Sciences*, vol. 68, pp. 695–727, Sept. 2021.

[27] P. S. Palar, T. Tsuchiya, and G. T. Parks, "Multi-fidelity non-intrusive polynomial chaos based on regression," *Computer Methods in Applied Mechanics and Engineering*, vol. 305, pp. 579–606, June 2016.

[28] J. Man, Q. Zheng, L. Wu, and L. Zeng, "Adaptive multi-fidelity probabilistic collocation-based Kalman filter for subsurface flow data assimilation: numerical modeling and real-world experiment," *Stochastic Environmental Research and Risk Assessment*, vol. 34, pp. 1135–1146, Aug. 2020.

[29] A. Fossà, R. Armellin, E. Delande, M. Losacco, and F. Sanfedino, "Multifidelity orbit uncertainty propagation using taylor polynomials," 2022.

[30] A. Fossà, R. Armellin, E. Delande, M. Losacco, and F. Sanfedino, "A multifidelity approach to robust orbit determination," *Acta Astronautica*, vol. 214, pp. 277–292, 2024.

[31] Z. McLaughlin, B. Jones, and R. Zanetti, "Nonlinear filtering with intrusive polynomial chaos for satellite uncertainty quantification," in *AAS/AIAA Space Flight Mechanics Meeting*, (Austin, TX), January 2023. AAS 23-370 (16 pages).

[32] B. Sudret, "Global sensitivity analysis using polynomial chaos expansions," *Reliability Engineering & System Safety*, vol. 93, pp. 964–979, July 2008.

[33] 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, pp. 18–34, 2015.

[34] O. M. Knio and O. P. L. Maître, "Uncertainty propagation in cfd using polynomial chaos decomposition," *Fluid Dynamics Research*, vol. 38, p. 616, sep 2006.

[35] Z. Liu, D. Lesselier, B. Sudret, and J. Wiart, "Surrogate modeling based on resampled polynomial chaos expansions," *Reliability Engineering & System Safety*, vol. 202, p. 107008, 2020.

[36] D. Xiu, *Numerical Methods for Stochastic Computations*. Princeton: Princeton University Press, 2010.

[37] B. J. Debusschere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, and O. P. Le Maître, "Numerical challenges in the use of polynomial chaos representations for stochastic processes," *SIAM Journal on Scientific Computing*, vol. 26, no. 2, pp. 698–719, 2004.

[38] 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, pp. 1–1, 10 2020.

[39] "Chaospy — chaospy 4.3.13 documentation."

[40] B. Tapley, J. Ries, S. Bettadpur, D. Chambers, M. Cheng, F. Condi, and S. Poole, "The GGM03 Mean Earth Gravity Model from GRACE,"

[41] W. M. Folkner, J. G. Williams, D. H. Boggs, R. S. Park, and P. Kuchynka, "The Planetary and Lunar Ephemerides DE430 and DE431,"