A MAXIMUM LIKELIHOOD APPROACH TO RECURSIVE POLYNOMIAL CHAOS PARAMETER ESTIMATION

Benjamin L. Pence[1], Hosam K. Fathy[2], and Jeffrey L. Stein[3]

(1) Graduate Student, (2) Assistant Research Scientist, (3) Professor,
Mechanical Engineering Department, The University of Michigan, Ann Arbor, MI 48109-2125
(bpence, hfathy, stein)@umich.edu

Abstract

This paper presents a method for recursively estimating the static parameters of linear or nonlinear stochastic dynamic systems given the systems' inputs and outputs. The paper accomplishes this objective by combining polynomial chaos theory with maximum likelihood estimation. The parameter estimates are calculated in a recursive or iterative manner. To the best of the author's knowledge, this is the first paper to address recursive maximum likelihood parameter estimation using polynomial chaos theory. The proposed approach is demonstrated on two systems: a linear 2nd order system with unknown damping and natural frequency, and a nonlinear Van der Pol oscillator with an unknown nonlinear damping coefficient. Because this recursive estimator is applicable to nonlinear systems, the authors portend that this novel formulation will be useful for a broad range of estimation problems.

1. Introduction

This paper presents a novel recursive parameter estimation method that combines generalized polynomial chaos (gPC) theory [1] with maximum likelihood estimation. The approach of this paper is applicable to nonlinear dynamic systems, including systems in which the unknown parameters are nonlinear in the state equations. The proposed approach uses polynomial chaos expansions to separate the stochastic part from the time-dependent part of the dynamic equations. Then the time-dependent (dynamic) part of the equations can be solved in a deterministic manner via the Galerkin method or stochastic collocation. Given system observations, and assuming Gaussian observation noise with known covariance, the proposed approach recursively estimates the most likely values of the unknown parameters using maximum likelihood estimation theory.

Polynomial chaos methods can be used for both dynamic system modeling and estimation applications. In the modeling area, Sandu *et al.* proposed a method for representing multibody dynamic systems with uncertainties using polynomial chaos theory [2]. Because the dynamic equations are solved only once (with a higher dimensional state space), gPC theory often provides a more computationally efficient method than Monte Carlo techniques for solving stochastic dynamic equations.

In addition to stochastic dynamic system modeling, polynomial chaos theory has also been combined with the Kalman filter and its variants for state estimation. Blanchard *et al.* combined polynomial chaos theory with the extended Kalman filter for combined state and parameter estimation [3]. Li and Xiu proposed a gPC ensemble Kalman filter for improved estimation accuracy and computational efficiency [4]. Saad *et al.* proposed a gPC-based ensemble Kalman filter for system identification and monitoring [5], and Smith *et al.* combined gPC with the Luenberger observer for state estimation [6].

Polynomial chaos theory has also been used for parameter estimation - not combined with state estimation. Blanchard *et al.* proposed a Bayesian parameter estimator that selects parameter estimates based on the maximum *a posteriori* estimate [7]. This estimator calculates parameter estimates in a batch manner after all the data has been collected. Marzouk and Xiu [8] proposed a Bayesian approach to estimate parameters of systems governed by partial differential equations, and provided a study on the convergence of the polynomial chaos based estimators. This work used the stochastic collocation approach and extended earlier but similar work done by Marzouk *et al.* [9] which used the Galerkin method. Finally, Southward developed a framework for recursive parameter estimators based on gPC theory [10]. Southward's method used instantaneous gradients of quadratic cost functions to recursively calculate parameter estimates.

The approach proposed in this paper combines polynomial chaos theory with maximum likelihood estimation. The parameter estimates are calculated in a recursive or iterative manner. To the best of the author's knowledge, this is the first paper to address recursive

maximum likelihood parameter estimation using polynomial chaos theory.

The remainder of the paper is organized as follows: The next section introduces the method for solving stochastic dynamic equations using gPC theory. Section 3 derives the recursive parameter update law based on maximum likelihood estimation; this is the main contribution of the paper. Section 4 applies the estimator to two systems: a 2nd order oscillator and the nonlinear Van der Pol equation. Finally, Section 5 summarizes the paper's conclusions.

2. gPC Theory

This section presents a framework for solving stochastic differential equations using gPC theory. The gPC framework was developed by Xiu and Karniadakis [1] building off the groundbreaking work by Ghanem and Spanos [11] and the conceptualization by Wiener [12]. Its application to multibody dynamic systems was introduced by Sandu *et al.* [2]. This section follows the clear and concise gPC development presented by Li and Xiu [4].

A set of state equations, which may be nonlinear, is used to describe the dynamic behavior of a system.

$$\dot{x} = F(t, x, u; \theta), \qquad 0 \le t \le t_f \tag{1}$$

$$x(0) = x_0 \tag{2}$$

The vector $x \in \mathbb{R}^{n_s}$ contains the system states which are assumed to have known initial conditions $x_0 \in \mathbb{R}^{n_s}$, and the vector $\theta = \begin{bmatrix}\theta_1 & \theta_2 & \cdots & \theta_{n_p}\end{bmatrix}$ contains the unknown parameters. The input vector $u \in \mathbb{R}^{n_u}$ is known and time-varying. The "dot" notation signifies the derivative with respect to time t.

In general, observations on a system may be governed by a nonlinear output model $y = H(t, x, u; \theta)$. However, the scope of this paper is limited to systems having observations described by a linear, time-invariant model:

$$y = C(\theta)x \tag{3}$$

The output vector $y \in \mathbb{R}^{n_y}$ contains the system observations.

The unknown parameters are viewed as being functions of random variables, *i.e.*, $\theta_i = \theta_i(\xi_i)$ for $i = 1, \dots, n_p$. If the random variables are assumed to be independent, the joint density function is $\rho(\xi) = \prod_{i=1}^{n_p} \rho_i(\xi_i)$ where $\rho_i(\xi_i)$ is the distribution of the i^{th} random variable ξ_i, and $\xi = \begin{bmatrix}\xi_1 & \xi_2 & \cdots & \xi_{n_p}\end{bmatrix}$. Parametric uncertainty leads to uncertainty in the system states. Therefore, $x(t) = x(t, \xi)$ is also a function of the random variables ξ.

Following the gPC method, the unknown parameters $\theta(\xi)$ and system states $x(t, \xi)$ are expanded onto a basis of orthogonal polynomial functions. The choice of the polynomial basis functions depends on the assumed prior distribution. Often, limited information is known about the parameter prior distribution. Sandu *et al.* [2] suggested using the Beta distribution to describe the prior distribution of the unknown parameters of mechanical systems. The corresponding selection of polynomial basis functions is the set of Jacobi polynomials. A special case of the Beta distribution, the uniform distribution, corresponds to the Legendre polynomials. Xiu and Karniadakis related a group of prior distributions to the Askey-scheme polynomials, and the interested reader should consult their work [1].

Assuming the random variables ξ to be identically distributed, the unknown parameters are expanded as follows:

$$\theta_i(\xi_i) \approx \sum_{k=0}^{N_p} \theta_{i,k}\, \phi_k(\xi_i), \qquad i = 1, \dots n_p. \tag{4}$$

The expansion coefficients $\theta_{i,k}$ are chosen such that (4) is distributed according to the parameter prior distribution $\rho_i(\theta_i)$, and hence, $\theta_{i,k}$ are known for all i, k. The polynomials $\phi_k(\xi_i)$ are orthogonal with respect to the following inner product:

$$\langle \phi_k(\xi_i), \phi_j(\xi_i) \rangle_i \coloneqq \int \phi_k(\xi_i)\phi_j(\xi_i)\rho_i(\xi_i)d\xi_i \tag{5}$$

The system states are also expanded in terms of orthogonal polynomials. The q^{th} state is expanded as:

$$\hat{x}_q(t, \xi) \coloneqq \sum_{|\alpha|=0}^{N} x_{q,\alpha}(t)\Phi_\alpha(\xi), \quad q = 1, \dots, n_s \tag{6}$$

Here, the vector $\alpha \coloneqq (\alpha_1, \dots, \alpha_{n_p}) \in \mathbb{N}_0^{n_p}$ is an n_p-dimensional multi-index, and $|\alpha|$ is the sum of the vector elements, *i.e.* $|\alpha| \coloneqq \alpha_1 + \cdots + \alpha_{n_p}$. The expansion $\hat{x}_q(t, \xi)$ is the closest approximation of the true state $x_q(t, \xi)$ in the space spanned by the orthogonal polynomials $\Phi_\alpha(\xi)$. The n_p-variate polynomials $\Phi_\alpha(\xi)$ are products of the univariate polynomials $\phi_{\alpha_i}(\xi_i)$.

$$\Phi_\alpha(\xi) = \prod_{i=1}^{n_p} \phi_{\alpha_i}(\xi_i), \qquad |\alpha| \le N \tag{7}$$

These polynomials $\Phi_\alpha(\xi)$ are orthogonal [2] with respect to the inner product:

$$\langle \Phi_\alpha(\xi), \Phi_\beta(\xi) \rangle \coloneqq \int \Phi_\alpha(\xi)\Phi_\beta(\xi)\, \rho(\xi)d\xi \tag{8}$$

The total number of state-expansion coefficients $x_{q,\alpha}(t)$ and polynomial basis functions $\Phi_\alpha(\xi)$ per state is

$$r \coloneqq \frac{(N + n_p)!}{N!\, n_p!}. \tag{9}$$

This number grows rapidly as the polynomial order and the number of unknown parameters increase.

If the solution $x_q(t, \xi)$ is known, the expansion coefficients $x_{q,\alpha}(t)$ in (6) are chosen such that the error between $x_q(t, \xi)$ and $\hat{x}_q(t, \xi)$ is orthogonal to the basis functions Φ_α, *i.e.*, $\langle \hat{x}_q(t, \xi) - x_q(t, \xi), \Phi_\alpha(\xi) \rangle = 0$. Then

$\langle \hat{x}_q(t,\xi), \Phi_\alpha(\xi)\rangle = \langle x_q(t,\xi), \Phi_\alpha(\xi)\rangle$, and the state expansion coefficients $x_{q,\alpha}(t)$ satisfy the following:

$$x_{q,\alpha}(t) = \frac{\langle x_q(t,\xi), \Phi_\alpha(\xi)\rangle}{\langle \Phi_\alpha(\xi), \Phi_\alpha(\xi)\rangle} = \frac{\langle \hat{x}_q(t,\xi), \Phi_\alpha(\xi)\rangle}{\langle \Phi_\alpha(\xi), \Phi_\alpha(\xi)\rangle} \quad (10)$$
$$q = 1, \dots, n_s, \qquad |\alpha| \le N$$

Since the solution $x_q(t,\xi)$ or $\hat{x}_q(t,\xi)$ is generally not available, the gPC theory uses either the stochastic Galerkin method or the stochastic collocation method to determine the state-expansion coefficients $x_{q,\alpha}(t)$.

Note that the inner product defined in (8) can be viewed as an expectation operator, *i.e.*,

$$\mathbb{E}[\Phi_\alpha(\xi)\Phi_\beta(\xi)] = \langle \Phi_\alpha(\xi), \Phi_\beta(\xi)\rangle \quad (11)$$

Then, the Galerkin method seeks estimates $\hat{x}_{q,\alpha}(t)$ of the state-expansion coefficients $x_{q,\alpha}(t)$ by solving (1) and (2) in the following weak form:

$$\mathbb{E}\big[\dot{\hat{x}}(t,\xi)\Phi_\alpha(\xi)\big] = \mathbb{E}\big[F\big(t,\hat{x}(t,\xi);\theta(\xi)\big)\Phi_\alpha(\xi)\big], \qquad |\alpha| \le N \quad (12)$$
$$0 \le t \le t_f, \qquad \mathbb{E}[x(0)\Phi_\alpha(\xi)] = \mathbb{E}[x_0\Phi_\alpha(\xi)]$$

This results in a set of deterministic state equations having the estimated state-expansion coefficients $\hat{x}_{q,\alpha}(t)$ as the new state variables. These new deterministic state equations can be solved using numerical integration.

The stochastic collocation method is discussed next. It is often more straightforward to implement, especially for nonlinear systems [2]. However, it is generally less accurate than the Galerkin method [4]. Following the stochastic collocation method, a deterministic set of collocation points $\mu^{(1)}, \dots, \mu^{(Q)}$, $Q \ge r$ and $\mu^{(i)} \in \mathbb{R}^{n_p}$, is drawn from the parameter prior distribution $\rho(\xi)$. These deterministic collocation points $\mu^{(i)}$ are substituted for the random variables ξ in (1) and (2), *i.e.*,

$$\dot{z}^{(i)} = F\big(t, z^{(i)}(t); \theta(\mu^{(i)})\big), \qquad i = 1, \dots, Q \quad (13)$$
$$z^{(i)}(0) = z_0^{(i)} \quad (14)$$

Here, $z_q^{(i)} \coloneqq \sum_{|\alpha|=0}^{N} x_{q,\alpha}(t)\Phi_\alpha\big(\mu^{(i)}\big)$ is the q^{th} deterministic state of the i^{th} set of deterministic state equations. The resulting Q uncoupled sets of state equations (each set having n_s states) can be solved using numerical integration. Stacking the new sets of states $z^{(i)}$ into a column vector $Z \in \mathbb{R}^{Q\cdot n_s}$ yields:

$$Z \coloneqq \begin{bmatrix} z^{(1)} \\ \vdots \\ z^{(Q)} \end{bmatrix} \quad (15)$$

Note that the state-expansion of (6) can be written as the Euclidean inner product between two vectors:

$$\hat{x}_q(t,\xi) = P(\xi)^T X_q(t) \quad (16)$$

In this equation, if $\Phi_\alpha(\xi)$ is the k^{th} element of $P(\xi) \in \mathbb{R}^r$, then $\hat{x}_{q,\alpha}(t)$ is the k^{th} element of $X_q(t) \in \mathbb{R}^r$. Then the state-vector $\hat{x}(t,\xi)$ can be written in terms of (16) as

$$\hat{x}(t,\xi) = \mathbb{P}(\xi)X(t). \quad (17)$$

Here, $\mathbb{P}(\xi)$ and $X(t)$, are defined respectively as

$$\mathbb{P}(\xi) \coloneqq \begin{bmatrix} P(\xi)^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & P(\xi)^T \end{bmatrix}, \quad (18)$$

$$X(t) \coloneqq \begin{bmatrix} X_1(t) \\ \vdots \\ X_{n_s}(t) \end{bmatrix} \quad (19)$$

The matrix $\mathbb{P}(\xi) \in \mathbb{R}^{n_s \times r\cdot n_s}$ has dimensions determined by the number of original system states n_s and the number of elements in the vector $X(t) \in \mathbb{R}^{r\cdot n_s}$. Then Z from (15) can be written:

$$Z \coloneqq \begin{bmatrix} z^{(1)} \\ \vdots \\ z^{(Q)} \end{bmatrix} = \boldsymbol{A}X(t) \quad (20)$$

Here, the collocation matrix $\boldsymbol{A} \in \mathbb{R}^{Q\cdot n_s \times r\cdot n_s}$ is formed by substituting the collocation points $\mu^{(1)}, \dots, \mu^{(Q)}$ into $\mathbb{P}(\xi)$ and then stacking the resulting matrices:

$$\boldsymbol{A} \coloneqq \begin{bmatrix} \mathbb{P}\big(\mu^{(1)}\big) \\ \vdots \\ \mathbb{P}\big(\mu^{(Q)}\big) \end{bmatrix} \quad (21)$$

The vector $X(t)$, containing the state-expansion coefficients $\hat{x}_{q,\alpha}(t)$, is calculated by

$$X(t) = \boldsymbol{A}^{\#}Z. \quad (22)$$

The matrix $\boldsymbol{A}^{\#} \in \mathbb{R}^{r\cdot n_s \times Q\cdot n_s}$ is the Moore-Penrose pseudo inverse:

$$\boldsymbol{A}^{\#} \coloneqq (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \quad (23)$$

The collocation points $\mu^{(i)}$ must be selected such that $\boldsymbol{A}^{\#}$ exists. Finally,

$$X_{q+|\alpha|} = \hat{x}_{q,\alpha}(t). \quad (24)$$

Sandu *et al.* [2] provide a note on the relationship between the stochastic collocation and stochastic Galerkin methods, and they also suggest methods for implementing the Galerkin method on nonlinear systems.

3. Recursive Parameter Estimation

This section derives the recursive parameter update law for estimating the most likely values of the random variables ξ given the system observations. The estimates of the unknown parameters $\theta(\xi)$ are then calculated using (4).

This development assumes that the noise in the system output observations is zero mean and Gaussian with known covariance matrix $R \in \mathbb{R}^{n_y \times n_y}$. It also assumes that the system observations are mutually independent. It assumes the only uncertainty in (1) and (2) is due to the unknown parameters. Finally it assumes that the polynomial chaos approximations in (4) and (6) are exact. (This last assumption is satisfied for second order processes as the number of expansion terms goes to infinity [11]. In practice, the expansion must be truncated after a finite number of terms, and thus the parameter estimates via this method will only approximately satisfy the maximum likelihood

criterion.) Under these assumptions, the likelihood function becomes [13, 8]:

$$\mathcal{L}(\xi|y_{0:t}) = \prod_{\tau=0}^{t} \rho(y(\tau)|\xi) \propto exp\left\{-\frac{1}{2}\sum_{\tau=0}^{t}\big(y(\tau) - \hat{y}(\tau,\xi)\big)^T R^{-1}\big(y(\tau) - \hat{y}(\tau,\xi)\big)\right\} \quad (25)$$

Here, $\mathcal{L}(\xi|y_{0:t})$ is the likelihood function of the unknown parameters ξ conditioned on a matrix $y_{0:t}$ which contains all of the observations up to the current time t. The function $\rho(y(t)|\xi)$ is the probability density function of the observation $y(t)$ at time t conditioned on the parameters ξ; and $\hat{y}(t,\xi)$ is the output of the stochastic model, *i.e.*

$$\hat{y} \coloneqq C\big(\theta(\xi)\big)\hat{x}(t,\xi). \quad (26)$$

The maximum likelihood estimate $\hat{\xi}$ is the value of ξ that maximizes the likelihood function (25). Equation (25) is maximized when the magnitude of the negative term in the exponent is minimized:

$$J(t,\xi) \coloneqq \frac{1}{2}\sum_{\tau=0}^{t}\big(y(\tau) - \hat{y}(\tau,\xi)\big)^T R^{-1}\big(y(\tau) - \hat{y}(\tau,\xi)\big) \quad (27)$$

Thus the most likely value of ξ is:

$$\hat{\xi} = argmin\, J(t,\xi) \quad (28)$$

The ability to update $J(t,\xi)$ iteratively is critical to making the approach of this paper recursive. This paper leverages the benefits of polynomial chaos to separate the time and unknown parameter parts of the equation to make this recursion possible. By substituting (17) and (26) into (27) and performing a few algebraic manipulations, Equation (27) can be written as:

$$J(t,\xi) \coloneqq \frac{1}{2}\sum_{i=1}^{n_y}\sum_{j=1}^{n_y}[R^{-1}]_{i,j}\left(\left(\sum_{\tau=0}^{t} y_i y_j\right) - 2C_i\mathbb{P}\left(\sum_{\tau=0}^{t} Xy_j\right) + C_i\mathbb{P}\left(\sum_{\tau=0}^{t} XX^T\right)\big(C_j\mathbb{P}\big)^T\right) \quad (29)$$

Here $[R^{-1}]_{i,j}$ is the j^{th} element in the i^{th} row of the inverse covariance matrix R^{-1}. The scalar y_k is the k^{th} element of the observation vector y and C_k is the k^{th} row of the output matrix C. In (29) the independent variables are dropped to reduce notational complexity.

The key purpose in writing (27) as shown in (29) is to explicitly show that only the deterministic time-dependent parts, X and y, appear inside the time summations. The parts that depend on the unknown parameters, C and $\mathbb{P}$, appear outside of the time summations. This is a result of using polynomial chaos expansions. Because the time summations are deterministic, they can be updated recursively as will be shown next. Thus, $J(t,\xi)$ can be evaluated recursively.

Consider an arbitrary matrix $G(t)$ and define the matrix $D_G^{(t)}$:

$$D_G^{(t)} \coloneqq \sum_{\tau=0}^{t} G(\tau). \quad (30)$$

Then $D_G^{(t+\Delta t)} \coloneqq \sum_{\tau=0}^{t+\Delta t} G(\tau)$ (where Δt is the time between samples) can be determined using only knowledge of the matrices $D_G^{(t)}$ and $G(t+\Delta t)$ as follows:

$$D_G^{(t+\Delta t)} = D_G^{(t)} + G(t+\Delta t). \quad (31)$$

Even as time progresses, the dimensions of the matrix $D_G^{(t)}$ are fixed. Thus to update the time summations in (29), the summation matrices $D_{y_i y_j}^{(t)} \in \mathbb{R}$, $D_{Xy_j}^{(t)} \in \mathbb{R}^{r\cdot n_s}$, and $D_{XX^T}^{(t)} \in \mathbb{R}^{r\cdot n_s \times r\cdot n_s}$ are stored in memory and updated according to (31) at the next time step. Equation (29) can be written in terms of (30) as:

$$J(t,\xi) \coloneqq \frac{1}{2}\sum_{i=1}^{n_y}\sum_{j=1}^{n_y}[R^{-1}]_{i,j}\left(D_{y_i y_j}^{(t)} - 2C_i\mathbb{P}D_{Xy_j}^{(t)} + C_i\mathbb{P}D_{XX^T}^{(t)}\big(C_j\mathbb{P}\big)^T\right) \quad (32)$$

The forgoing discussion has outlined a procedure for determining recursively the term in the exponent of the likelihood function (25). The remaining challenge is to determine the value of ξ that minimizes the term in the exponent, $J(t,\xi)$, thus maximizing the likelihood at each time step. This can be viewed as an optimization problem in which the objective function $J(t,\xi)$ is time-varying. The following sections offer potential solution approaches.

3.1 Solution via Gradient Descent

Proceeding in a manner similar to and indeed inspired by Southward's [10], this paper proposes a gradient based parameter update law.

$$\hat{\xi}_{t+1} = \hat{\xi}_t - \Gamma\frac{\partial J(t,\xi)}{\partial \xi}\bigg|_{\xi=\hat{\xi}} \quad (33)$$

Here $\hat{\xi}_t$ is the estimate of ξ at time t, and Γ is a user-specified gain matrix that can be chosen to vary in time. In static optimization, if Γ is the identity matrix, (33) is a steepest descent method. If Γ is the inverse Hessian matrix (matrix of second derivatives), then (33) is Newton's method, and if Γ is proportional to the Hessian matrix, then (33) is a modified Newton's method. These static optimization concepts may be helpful for selecting Γ.

Substituting (32) into (33) and using the fact that the time-dependent parts $D_{y_i y_j}^{(t)}$, $D_{Xy_j}^{(t)}$, and $D_{XX^T}^{(t)}$ can be moved

outside of the partial derivatives since they don't depend on the unknown parameters gives the following update law:

$$\begin{aligned}\hat{\xi}_{t+1} &= \hat{\xi}_t - \Gamma \sum_{i=1}^{n_y}\sum_{j=1}^{n_y}[R^{-1}]_{i,j} \\ &\times \left(\left.\frac{\partial C_i \mathbb{P}}{\partial \xi}\right|_{\xi=\hat{\xi}} \left(\left. D_{XX^T}^{(t)} \left(C_j \mathbb{P}\right)^T\right|_{\xi=\hat{\xi}} - D_{Xy_j}^{(t)}\right)\right)\end{aligned} \tag{34}$$

The magnitude of the values in the time summations $D_{XX^T}^{(t)}$ and $D_{Xy_j}^{(t)}$ may grow unbounded in time, so normalization may be necessary. Note that the magnitudes $\left\| D_{XX^T}^{(t)}[\mathbb{P}^T C_j{}^T]_{\xi=\hat{\xi}} \right\|_2 = \left\| \sum_{\tau=0}^{t} X\hat{y}_j \right\|_2$ and $\left\| \sum_{\tau=0}^{t} X y_j \right\|_2 = \left\| D_{Xy_j}^{(t)} \right\|_2$, therefore normalizing (34) by the scalar $\left(\left\| D_{Xy_j}^{(t)} \right\|_2 \right)^{-1}$ is a judicious choice.

In practice, to avoid division by zero, (34) is normalized by $\left(1 + \left\| D_{Xy_j}^{(t)} \right\|_2\right)^{-1}$. Then the final update law becomes:

$$\begin{aligned}\hat{\xi}_{t+1} = \hat{\xi}_t - \Gamma \sum_{i=1}^{n_y}\sum_{j=1}^{n_y}[R^{-1}]_{i,j} \Bigg(&\left.\frac{\partial C_i \mathbb{P}}{\partial \xi}\right|_{\xi=\hat{\xi}} \\ &\times \frac{\left(\left. D_{XX^T}^{(t)} \left(C_j \mathbb{P}\right)^T\right|_{\xi=\hat{\xi}} - D_{Xy_j}^{(t)}\right)}{\left(1 + \left\| D_{Xy_j}^{(t)} \right\|_2\right)} \Bigg)\end{aligned} \tag{35}$$

The normalization scalar $\left(1 + \left\| D_{Xy_j}^{(t)} \right\|_2\right)^{-1}$ is not a function of the random variable ξ and therefore (35) still seeks to satisfy the maximum likelihood criterion (25).

The partial derivative $\left.\frac{\partial (C_j \mathbb{P})}{\partial \xi_i}\right|_{\xi=\hat{\xi}}$ in (35) can be expanded using the product rule:

$$\left.\frac{\partial \left(C_j \mathbb{P}\right)}{\partial \xi_i}\right|_{\xi=\hat{\xi}} = \left.\frac{\partial \left(C_j\right)}{\partial \xi_i}\mathbb{P}\right|_{\xi=\hat{\xi}} + C_j \left.\frac{\partial (\mathbb{P})}{\partial \xi_i}\right|_{\xi=\hat{\xi}} \tag{36}$$

Shimp offers a clever method for calculating the partial derivatives $\left.\frac{\partial (\mathbb{P})}{\partial \xi_i}\right|_{\xi=\hat{\xi}}$, and the interested reader should consult his work [14].

3.2 Solution via Random Search

The gradient descent solution of Section 3.1 doesn't guarantee that the estimated parameters will *globally* minimize $J(t,\xi)$, but may potentially select parameters satisfying only *local* minima. This subsection proposes a strategy for enabling the parameter update law to escape local minima in order to satisfy the maximum likelihood criterion.

The function $J(t,\xi)$ can be evaluated at time t for any realization of ξ. This suggests the following strategy:

Let $\mathbb{N} \coloneqq \{1,2,3,\dots\}$ be the set of natural numbers. At each time instant, select $n_r \in \mathbb{N}$ realizations of ξ randomly, and evaluate the cost of each realization using (32). Then compare these costs with the cost of the previous parameter estimate $\hat{\xi}_{t-1}$, and set the new estimate $\hat{\xi}_t$ to be the realization with the lowest cost. This random search strategy allows the algorithm to search any point in the entire parameter space and thus escape local minima.

A guided random search policy combines the random search with the gradient search: The cost of the gradient solution from (35) is compared with the costs of the n_r randomly selected realizations as well as the cost of the previous estimate $\hat{\xi}_{t-1}$. The new estimate is chosen to be the realization with the lowest cost.

4. Examples

This section presents two examples to demonstrate the recursive parameter estimator proposed in this paper. The first example considers estimating the damping and natural frequency of a forced second order oscillator. It uses the Galerkin approach. The second example uses the collocation approach and seeks to estimate a parameter of the Van der Pol equation.

4.1 Forced Second Order Oscillator

Consider a forced, second order, stochastic differential equation with zero initial conditions. The damping and natural frequency, ζ and ω, are independent, uncertain parameters. The scalar system input u is a normally distributed random sequence with zero mean and unit variance.

$$\begin{aligned}\ddot{x} + 2\zeta\omega\dot{x} + \omega^2 x &= u \\ y &= -2\zeta\omega\dot{x} - \omega^2 x \\ \zeta &\sim \mathcal{N}\left(\mu_\zeta, \sigma_\zeta{}^2\right) \\ \omega &\sim \mathcal{N}\left(\mu_\omega, \sigma_\omega{}^2\right)\end{aligned} \tag{37}$$

Equation (37) assumes the uncertain parameters are from the Gaussian distribution; therefore, the Hermite polynomials form the basis functions [1]. The first few Hermite polynomials of variable ξ_i are shown here:

$$\begin{aligned}H_0(\xi_i) &= 1 \\ H_1(\xi_i) &= 2\xi_i \\ H_2(\xi_i) &= 4\xi_i{}^2 - 2 \\ H_3(\xi_i) &= 8\xi_i{}^3 - 12\xi_i \\ H_4(\xi_i) &= 16\xi_i{}^4 - 48\xi_i{}^2 + 12 \\ H_5(\xi_i) &= 32\xi_i{}^5 - 160\xi_i{}^3 + 120\xi_i\end{aligned} \tag{38}$$

Let ξ_1 and ξ_2 be independent standard Gaussian variables. Since $\xi_i \sim \mathcal{N}(0,1)$, and by properties of Gaussian random variables, the damping and natural frequency of the system of Equation (37) can be written as follows:

$$\begin{aligned}\zeta(\xi_1) &= \mu_\zeta + \sigma_\zeta \xi_1 \\ \omega(\xi_2) &= \mu_\omega + \sigma_\omega \xi_2\end{aligned} \tag{39}$$

The system state and output equations are expanded in terms of the polynomial basis functions:

$$\begin{bmatrix}\dot{\hat{x}}_1(t,\xi_1,\xi_2)\\ \dot{\hat{x}}_2(t,\xi_1,\xi_2)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega(\xi_2)^2 & -2\zeta(\xi_1)\omega(\xi_2)\end{bmatrix}\begin{bmatrix}\hat{x}_1(t,\xi_1,\xi_2)\\ \hat{x}_2(t,\xi_1,\xi_2)\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}u(t) \quad (40)$$

$$\hat{y}(t,\xi_1,\xi_2) = C(\xi_1,\xi_2)\begin{bmatrix}\hat{x}_1(t,\xi_1,\xi_2)\\ \hat{x}_2(t,\xi_1,\xi_2)\end{bmatrix}$$

with

$$C(\xi_1,\xi_2) = [-\omega(\xi_2)^2 \quad -2\zeta(\xi_1)\omega(\xi_2)] \quad (41)$$

Where the q^{th} state is expanded as in (6):

$$\hat{x}_q(t,\xi) = \sum_{|\alpha|=0}^{N} x_{q,\alpha}(t)\Phi_\alpha(\xi) = \sum_{i=0}^{N}\sum_{j=0}^{N-i}\hat{x}_{q,(i,j)}(t)H_i(\xi_1)H_j(\xi_2), \quad q = 1,\dots,n_s \quad (42)$$

Following the Galerkin method (12), the expanded states are projected onto the basis functions $\Phi_\alpha(\xi)$. We define the following matrices:

$$\Delta_{\langle\Phi,\Phi\rangle} := \begin{bmatrix}\langle\Phi_0,\Phi_0\rangle & \langle\Phi_0,\Phi_1\rangle & \cdots & \langle\Phi_0,\Phi_\sigma\rangle\\ \langle\Phi_1,\Phi_0\rangle & \langle\Phi_1,\Phi_1\rangle & \cdots & \langle\Phi_1,\Phi_\sigma\rangle\\ \vdots & & \ddots & \vdots\\ \langle\Phi_\sigma,\Phi_0\rangle & \langle\Phi_\sigma,\Phi_1\rangle & \cdots & \langle\Phi_\sigma,\Phi_\sigma\rangle\end{bmatrix} \quad (43)$$

$$\Delta_{\langle\Phi,Q\Phi\rangle} := \begin{bmatrix}\langle\Phi_0,Q\Phi_0\rangle & \langle\Phi_0,Q\Phi_1\rangle & \cdots & \langle\Phi_0,Q\Phi_\sigma\rangle\\ \langle\Phi_1,Q\Phi_0\rangle & \langle\Phi_1,Q\Phi_1\rangle & \cdots & \langle\Phi_1,Q\Phi_\sigma\rangle\\ \vdots & & \ddots & \vdots\\ \langle\Phi_\sigma,Q\Phi_0\rangle & \langle\Phi_\sigma,Q\Phi_1\rangle & \cdots & \langle\Phi_\sigma,Q\Phi_\sigma\rangle\end{bmatrix} \quad (42)$$

$$Q = \xi_1,\ \xi_2,\ \xi_1\xi_2, or\ {\xi_2}^2$$

and

$$B_2 := \begin{bmatrix}\pi\\0\\\vdots\\0\end{bmatrix} \quad (45)$$

Then the resulting deterministic equations are

$$\dot{X} = \begin{bmatrix}\mathbf{0} & I_R\\ A_{21} & A_{22}\end{bmatrix}X + \begin{bmatrix}\mathbf{0}\\B_G\end{bmatrix}u(t) \quad (46)$$

where

$$A_{21} := -{\mu_\omega}^2 I_{\sigma+1} - 2\mu_\omega\sigma_\omega{\Delta_{\langle\Phi,\Phi\rangle}}^{-1}\Delta_{\langle\Phi,\xi_2\Phi\rangle} - {\sigma_\omega}^2{\Delta_{\langle\Phi,\Phi\rangle}}^{-1}\Delta_{\langle\Phi,{\xi_2}^2\Phi\rangle} \quad (47)$$

$$A_{22} := -2\big(\mu_\omega\mu_\zeta I_{\sigma+1} + \mu_\omega\sigma_\zeta{\Delta_{\langle\Phi,\Phi\rangle}}^{-1}\Delta_{\langle\Phi,\xi_1\Phi\rangle} + \mu_\zeta\sigma_\omega{\Delta_{\langle\Phi,\Phi\rangle}}^{-1}\Delta_{\langle\Phi,\xi_2\Phi\rangle} + \sigma_\zeta\sigma_\omega{\Delta_{\langle\Phi,\Phi\rangle}}^{-1}\Delta_{\langle\Phi,\xi_1\xi_2\Phi\rangle}\big) \quad (48)$$

and

$$B_G \stackrel{\text{def}}{=} {\Delta_{\langle\Phi,\Phi\rangle}}^{-1}B_2. \quad (49)$$

To use the update law from (35), calculate $\mathbb{P}(\hat{\xi})$, $\frac{\partial(\mathbb{P})}{\partial\xi_i}\Big|_{\xi=\hat{\xi}}$, $C(\hat{\xi})$, $\frac{\partial(C_j)}{\partial\xi_i}\Big|_{\xi=\hat{\xi}}$, and X. $\mathbb{P}(\boldsymbol{\xi})\epsilon\mathbb{R}^{2\times 2r}$ is calculated using Equation (18) where:

$$P = \begin{bmatrix}\Phi_{(0,0)} & \Phi_{(0,1)} & \cdots & \Phi_{(N,0)}\end{bmatrix}^T\epsilon\mathbb{R}^r \quad (52)$$

and $\Phi_{(i,j)} = H_i(\xi_1)H_j(\xi_2)$ for $i,j = 0,1,\dots,N$ can be calculated using the Hermite polynomials (38). Then,

$$\frac{\partial\mathbb{P}(\boldsymbol{\xi})}{\partial\xi_i} = \begin{bmatrix}\frac{\partial P}{\partial\xi_i} & \mathbf{0}\\ \mathbf{0} & \frac{\partial P}{\partial\xi_i}\end{bmatrix}; i = 1,2 \quad (53)$$

The partial derivative $\frac{\partial P}{\partial\xi_i}$ is straightforward because each $H_l(\xi_i)$ for $l = 0,1,\dots,N$ is a polynomial in ξ_i with known constant coefficients.

This section presents simulation runs which for simplicity assumed the observation noise variance to be $R = 1$ which was different than the true simulated observation noise variance $R = 0.015$. The gain matrix was chosen as $\Gamma = 0.75\cdot I_{2\times 2}$. Zero mean Gaussian process noise with variance 0.015 was also added to the input $u(t)$.

Figure 1 shows the convergence of the proposed algorithm under the conditions described above for different initial guesses of μ_ζ and μ_ω. The initial guesses were chosen so that the system (46) is stable. In Run 1, $\mu_\zeta = 0.3$ and $\mu_\omega = 13$. In Run 2, $\mu_\zeta = 0.3$ and $\mu_\omega = 6$. In run3, $\mu_\zeta = 0.9$ and $\mu_\omega = 6$. And in Run 4, $\mu_\zeta = 0.9$ and $\mu_\omega = 13$. In each case, $\sigma_\zeta = 0.15$ and $\sigma_\omega = 2$. The simulation step size was 0.005 seconds. The true values of ζ and ω were 0.5 (unitless) and 10 radians per second respectively.

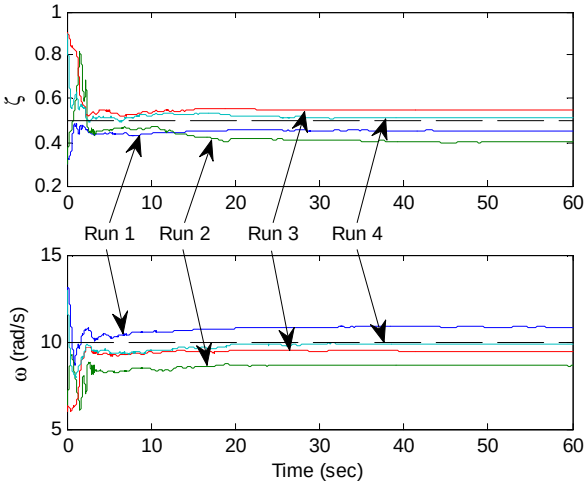

Fig. 1: Convergence of the algorithm for different initial conditions.

In the simulations, the standard deviation of the observation noise was 0.12, and the standard deviation of the signal was 0.215. Thus the signal-to-noise ratio was 5dB. Despite the large amount of signal noise, the algorithm converged fairly accurately as shown in Figure 1. The algorithm converged to steady state within about 20 seconds or equivalently within about 4,000 data points.

4.2 Van der Pol Oscillator

Consider the forced nonlinear Van der Pol Equation with unknown parameter $\varepsilon \sim Uniform([5,11])$:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - \frac{1}{\varepsilon}({x_1}^2 - 1)x_2 + \sin(50t) \end{aligned} \tag{55}$$

Here, the initial conditions $x_1 = 0.1$ and $x_2 = -0.1$ are known. The true value of the unknown parameter is $\varepsilon^* = 10$. The first state is observed, *i.e.*,

$$y = x_1. \tag{56}$$

Because the unknown parameter prior distribution is uniform, the appropriate set of basis functions is the Legendre polynomials on the interval [-1,1]. The prior distribution of ε can be written in terms of the random variable $\xi \sim Uniform([-1,1])$ as:

$$\varepsilon = \varepsilon_0 + \varepsilon_1 \xi = 8 + 3\xi \tag{57}$$

Next, the states are expanded onto the polynomial basis functions:

$$\begin{aligned} \hat{x}_q(t,\xi) &= \sum_{|\alpha|=0}^{N} x_{q,\alpha}(t)\Phi_\alpha(\xi) \\ &= \sum_{i=0}^{N} x_{q,i}(t)\mathrm{L}_i(\xi), \qquad q = 1,2 \end{aligned} \tag{58}$$

Here, $\mathrm{L}_i(\xi)$, $i = 0, \dots, N$ are the Legendre polynomials of order up to N on the interval [-1,1].

Following the collocation method, a set of collocation points $\mu^{(1)}, \dots, \mu^{(Q)}$ are drawn from the $Uniform([-1,1])$ distribution. Then $\mathbb{P}(\mu^{(i)}) \epsilon \mathbb{R}^{2\times 2N}$, $i = 1, \dots, Q$ is formed using (18) where

$$P(\mu^{(i)})^T = \begin{bmatrix} \mathrm{L}_0(\mu^{(i)}) & \cdots & \mathrm{L}_N(\mu^{(i)}) \end{bmatrix} \tag{59}$$

and $\boldsymbol{A}^{\#}$ is calculated using (21-23). Then Q sets of equation (55) are run simultaneously, and the resulting state-trajectories $z^{(i)}$ are stacked into a column vector as in (15). Then, at each iteration, Equation (22) is used to find the vector $X(t)$.

To use the parameter estimator (35), we first calculate $\left[\mathbb{P}^T {C_j}^T\right]_{\xi=\hat{\xi}_t}$ and $\left.\frac{\partial(C_j\mathbb{P})}{\partial\xi}\right|_{\xi=\hat{\xi}_t}$. Since only the first state x_1 is observed, $C_j\mathbb{P} = P(\xi)^T$, and thus:

$$\left[\mathbb{P}^T {C_j}^T\right]_{\xi=\hat{\xi}_t} = [P(\xi)]_{\xi=\hat{\xi}_t} \tag{60}$$

and

$$\left.\frac{\partial(C_j\mathbb{P})}{\partial\xi_i}\right|_{\xi=\hat{\xi}_t} = \left.\frac{\partial(P(\xi)^T)}{\partial\xi}\right|_{\xi=\hat{\xi}_t} \tag{61}$$

Again the derivative $\frac{\partial(P(\xi)^T)}{\partial\xi}$ is straightforward as it requires derivatives of the polynomial functions $\mathrm{L}_i(\xi)$, $i = 0, \dots, N$.

In Figure 2, $N = 5$, $\Gamma = {}^1/_2$, and the time step was 0.1 seconds. Figure 5 shows the convergence of the proposed algorithm. The signal-to-noise ratio in the observations was varied between low noise, 15dB, and high noise, 5dB, values.

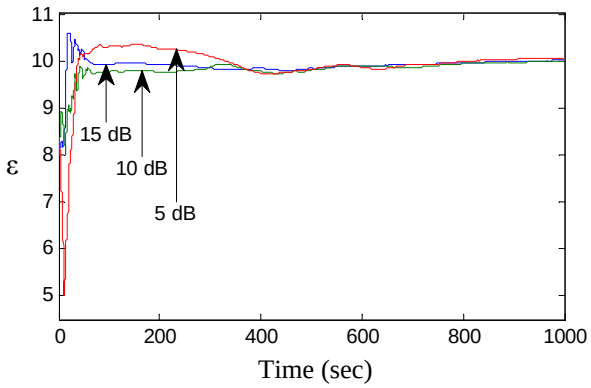

Fig. 2: Convergence of the algorithm for variable noise.

Under the conditions of this simulation study of the nonlinear oscillator, the proposed estimator appeared to be consistent; i.e, increasing the noise level changed the convergence rate, but the algorithm eventually converged to the same parameter value as time approached infinity.

5. Conclusions

This paper derived a maximum likelihood approach to recursive estimation based on polynomial chaos theory. It demonstrated the proposed algorithm on two systems: a linear, 2nd order, differential equation and the nonlinear Van der Pol equation. This paper discussed polynomial chaos theory using both the stochastic Galerkin and stochastic collocation methods and demonstrated each approach in the examples.

The proposed estimator has potential for parameter estimation in nonlinear systems, making it valuable to a wide range of parameter estimation problems.

Acknowledgments

This research was funded by the U.S. Army TARDEC through its center for excellence in automotive modeling and simulation. The authors would like to acknowledge this financial support.

References

[1] Xiu D., Kaniadakis G.E., (2002), "The Wiener-Askey polynomial chaos for stochastic differential equations", *SIAM J. on Scientific Computing*, V. 24, n 2, pp 619-44.

[2] Sandu, A., Sandu, C., Ahmadian, M., (2006), "Modeling multibody systems with uncertainties. Part 1: theoretical and computational aspects", *Multibody Syst. Dyn.* , v 15, n 4, p 373-95, May 2006

[3] Blanchard, E., Sandu, A., and Sandu, C., (2007), "Parameter Estimation Method Using an Extended Kalman Filter", Proc. ISTVS conference, Alaska, USA,

[4] Li, J. and Xiu, D., (2009), "A Generalized Polynomial Chaos Based Ensemble Kalman Filter with High Accuracy", Journal of Computational Physics, v. 228, pp. 5454-5694.

[5] Saad, G., Ghanem, R.G., and Masri, S., (2007), "Robust system identification of strongly non-linear dynamics using a polynomial chaos based sequential data assimilation technique", *Collection of Technical Papers - 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Vol. 6, pp. 6005-6013, Honolulu, HI.

[6] Smith, A. H. C.; Monti, A.; Ponci, F.; (2007), "Indirect Measurements via a Polynomial Chaos Observer" *IEEE Transactions on Instrumentation and Measurement*, **56**(3), pp. 743 – 752.

[7] Blanchard, E., Sandu, A., and Sandu, C., (2008), "A polynomial-chaos-based Bayesian approach for estimating uncertain parameters of mechanical systems", 19th International Conference on Design Theory and Methodology; 1st International Conference on Micro- and Nanosystems; and 9th International Conference on Advanced Vehicle Tire Technologies, Parts A and B; Vol. 3, pp. 1041 -1048

[8] Marzouk, Y. M., Xiu, D., (2009), "A stochastic collocation approach to Bayesian Inference in Inverse Problems", Communications in Computational Physics, Vol. 6, No. 4, pp. 826-847

[9] Marzouk, Y. M., Najm, N. N., Rahn, L. A., (2007), "Stochastic spectral methods for efficient Bayesian solution of inverse problems", Journal of Computational Physics, Vol. 224, pp. 560-586

[10] Southward, S.C. (2008), "Real-Time Parameter ID using Polynomial Chaos Expansions", *ASME International Mechanical Engineering Congress and Exposition, Proceedings*, Vol. 9, Part B, pp. 1167-1174

[11] Ghanem, R., Spanos, P., (1991), *Stochastic finite elements: A spectral approach*, Springer-Verlag New York Inc.

[12] Wiener, N., (1938), "The homogenous chaos", Amer. J. Math., 60 (1938), pp. 897-936.

[13] Moon, T.K. and Stirling, W.C., (2000), *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, pp. 542-567

[14] Shimp, S. K. III, (2008), "Vehicle Sprung Mass Identification Using an Adaptive Polynomial-Chaos Method", M.S. Thesis, *Virginia Polytechnic Institute and State University*