Polynomial Chaos

Raktim Bhattacharya

Intelligent Systems Research Laboratory
Aerospace Engineering, Texas A&M University.

isrlab.github.io

Introduction

What is polynomial chaos theory?

It provides a non-sampling based method to determine evolution of uncertainty in dynamical system, when there is probabilistic uncertainty in the system parameters.

Consider a dynamical system

- $\dot{x} = -ax$, $x(t_0) = x_0$ is given (known)
- a is an unknown parameter in the range $[0, 1]$ (equally likely values)

Polynomial chaos theory helps us answer these questions

- How does $x(t)$ evolve for various values of a?
- What is the ensemble behavior of x (mean, variance, PDF)?

Monte-Carlo Approach

Summary of Steps

- $\dot{x} = -ax,\ x(t_0) = 1$
- a is an unknown parameter in the range $[0, 1]$ (equally likely values)
- Sample $a \in [0, 1]$
- Plot $x(t)$ for every value of a
- Estimate statistics from data

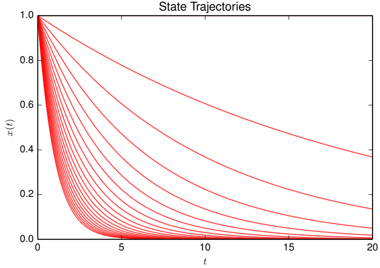
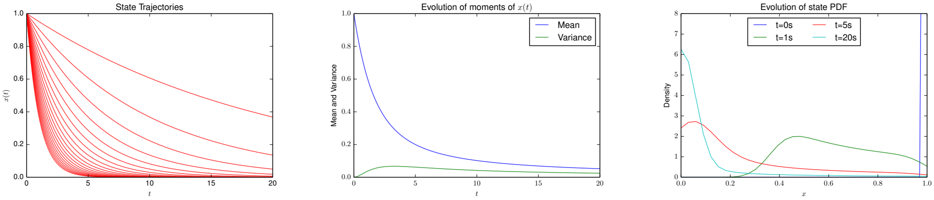


Figure: Sample paths

Monte-Carlo Approach

Solution Interpretation



(a) Sample paths (b) Moments (c) State PDF

Figure: Solution from Monte-Carlo simulations

- The solution $x(t)$ depends on parameter a
- More appropriate to write it as $x(t, a)$
- Solution statistics are time varying
- State PDF is also time varying
- Other methods to characterize $x(t, a)$?

Polynomial Chaos

Basic Idea

- Approximate $x(t,a)$, solution of $\dot{x} = -ax$ as

$$\hat{x}(t,a) \approx \sum_i x_i(t)\phi_i(a)$$

- $\phi_i(a)$ are known polynomials of parameter a
- $x_i(t)$ are unknown time varying coefficients
- Determine $x_i(t)$ that minimises equation error $e(t,a) = \dot{\hat{x}} - a\hat{x}$
  - Galerkin Projection: minimize $\|e(t,a)\|_2$
  - Stochastic Collocation: set $e(t,a) = 0$ at certain locations
- Resulting system
  - is in higher dimensional state space
  - doesn't involve parameter a

Galerkin Projection

Raktim Bhattacharya
Aerospace Engineering, Texas A&M University
isrlab.github.io

Stochastic Finite Element

Generalized Formulation

- Let system be

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\Delta}),$$

  where state $\boldsymbol{x} \in \mathbb{R}^n$ and parameter $\boldsymbol{\Delta} \in \mathcal{D}_{\boldsymbol{\Delta}} \subseteq \mathbb{R}^d$
  - More precisely, $\boldsymbol{\Delta} := \boldsymbol{\Delta}(\omega)$ is a $\mathbb{R}^d$-valued continuous random variable
  - ω is an event in the probability space $(\Omega, \mathcal{F}, P)$
- A second order process $\boldsymbol{x}(t, \boldsymbol{\Delta}(\omega))$ can be expressed by polynomial chaos as

$$\boldsymbol{x}(t, \boldsymbol{\Delta}(\omega)) = \sum_{i=0}^{\infty} \boldsymbol{x}_i(t)\phi_i(\boldsymbol{\Delta}(\omega))$$

- In practice, approximate with finite terms

$$\boldsymbol{x}(t, \boldsymbol{\Delta}) \approx \hat{\boldsymbol{x}}(t, \boldsymbol{\Delta}) = \sum_{i=0}^{N} \boldsymbol{x}_i(t)\phi_i(\boldsymbol{\Delta})$$

Reduced Order System

1. Dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\Delta}), \quad (n \text{ differential equations})$$

2. Proposed solution

$$\hat{\boldsymbol{x}}(t, \boldsymbol{\Delta}) = \sum_{i=0}^{N} \boldsymbol{x}_i(t)\phi_i(\boldsymbol{\Delta})$$

3. Residue

$$\boldsymbol{e}(t, \boldsymbol{\Delta}) := \dot{\hat{\boldsymbol{x}}} - \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{\Delta})$$

4. Set projection on basis function to zero (best $\mathcal{L}_2$ solution)

$$\langle \boldsymbol{e}(t, \boldsymbol{\Delta}), \phi_i(\boldsymbol{\Delta}) \rangle = 0, \text{ for } i = 0, 1, \cdots, N$$

5. This gives $n(N+1)$ ordinary differential equations to determine $n(N+1)$ unknowns $\boldsymbol{x}_i(t) \in \mathbb{R}^n$

Inner product

Define

$$\langle \boldsymbol{e}(t,\boldsymbol{\Delta}), \phi_i(\boldsymbol{\Delta})\rangle := \int_{\mathcal{D}_{\boldsymbol{\Delta}}} \boldsymbol{e}(t,\boldsymbol{\Delta})\phi_i(\boldsymbol{\Delta})p(\boldsymbol{\Delta})d\boldsymbol{\Delta},$$

where $p(\boldsymbol{\Delta})$ is the probability density function of $\boldsymbol{\Delta}$.

Also

$$\mathbf{E}\left[\boldsymbol{e}(t,\boldsymbol{\Delta})\phi_i(\boldsymbol{\Delta})\right] := \int_{\mathcal{D}_{\boldsymbol{\Delta}}} \boldsymbol{e}(t,\boldsymbol{\Delta})\phi_i(\boldsymbol{\Delta})p(\boldsymbol{\Delta})d\boldsymbol{\Delta}$$

Therefore,

$$\langle \boldsymbol{e}(t,\boldsymbol{\Delta}), \phi_i(\boldsymbol{\Delta})\rangle \equiv \mathbf{E}\left[\boldsymbol{e}(t,\boldsymbol{\Delta})\phi_i(\boldsymbol{\Delta})\right]$$

Basis Functions

Basis functions are such that

$$\mathbf{E}\left[\phi_i(\boldsymbol{\Delta})\phi_j(\boldsymbol{\Delta})\right] = 0, \text{ when } i \neq j$$

i.e. orthogonal w.r.t $p(\boldsymbol{\Delta})$

$$\int_{\mathcal{D}_{\boldsymbol{\Delta}}} \phi_i(\boldsymbol{\Delta})\phi_j(\boldsymbol{\Delta})p(\boldsymbol{\Delta})d\boldsymbol{\Delta} = 0, \text{ when } i \neq j$$

Distribution	**Polynomial Basis Function**	**Support**
Uniform: $\frac{1}{2}$	Legendre	$x \in [-1, 1]$
Standard Normal: $\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$	Hermite	$x \in (-\infty, \infty)$
Beta: $\frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1}$	Jacobi	$x \in [0, 1]$
Gamma: $\frac{x^{k-1}e^{-\frac{x}{\theta}}}{\theta^k\Gamma(k)}$	Laguerre	$x \in (0, \infty)$

Basis Functions (contd.)

In general

- $\{\phi_i(\boldsymbol{\Delta})\}$ are orthogonal polynomials with weight $p(\boldsymbol{\Delta})$
- $\mathcal{L}_2$ exponential convergence in corresponding Hilbert functional space
- Askey scheme of hypergeometric polynomials for common $p(\boldsymbol{\Delta})$
  - Normal, uniform, beta, gamma, etc
- Numerically generate for arbitrary $p(\boldsymbol{\Delta})$:
  - Gram-Schmidt
  - Chebyshev
  - Gauss-Wigert
  - Discretized Stieltjes

Basis Functions (contd.)

Mixed Basis Functions

- Let $\mathbf{\Delta} := [\Delta_1 \ \Delta_2]^T$, Δ_1 Δ_2 are independent
  - Δ_1 is uniform over $[-1, 1]$
  - Δ_2 is standard normal over $(-\infty, \infty)$
- What is the basis function for $\mathbf{\Delta}$?
- $\{\phi_i(\mathbf{\Delta})\}$ is multivariate polynomial
  - $\{\psi_j(\Delta_1)\}$: Legendre polynomials
  - $\{\theta_k(\Delta_2)\}$: Hermite polynomials
  - $\{\phi_i(\mathbf{\Delta})\}$: tensor product of $\{\psi_j(\Delta_1)\}$ and $\{\theta_k(\Delta_2)\}$

Example: First Order Linear System

Consider system $\dot{x} = -ax$, where $a \in \mathcal{U}_{[0,1]}$ (uniform distribution)

1. Define $a(\Delta) := \frac{1}{2}(1 + \Delta),\ \Delta \in \mathcal{U}_{[-1,1]}$

   Now dynamics is $\dot{x} = -a(\Delta)x$.

2. Approximate solution as $\hat{x} = \sum_{i=0}^{N} x_i(t)\phi_i(\Delta)$ (ϕ_i are Legendre polynomials)
3. Residue:

$$
\begin{aligned}
e(t,\Delta) &:= \dot{\hat{x}} - a(\Delta)\hat{x} \\
&= \sum_{i=0}^{N} \dot{x}_i(t)\phi_i(\Delta) - a(\Delta)\sum_{i=0}^{N} x_i(t)\phi_i(\Delta)
\end{aligned}
$$

Example: First Order Linear System (contd.)

4. Project residue on j^{th} basis function:

$$\langle e(t,\Delta), \phi_j(\Delta)\rangle = \left\langle \sum_{i=0}^{N} \dot{x}_i(t)\phi_i(\Delta), \phi_j(\Delta) \right\rangle - \left\langle a(\Delta)\sum_{i=0}^{N} x_i(t)\phi_i(\Delta), \phi_j(\Delta) \right\rangle$$
$$= \sum_{i=0}^{N} \dot{x}_i(t)\langle \phi_i(\Delta), \phi_j(\Delta)\rangle - \sum_{i=0}^{N} x_i(t)\langle a(\Delta)\phi_i(\Delta), \phi_j(\Delta)\rangle$$

5. If $\langle \phi_i(\Delta), \phi_j(\Delta)\rangle = 0$ for $i \neq j$ (orthogonal)

$$\langle e(t,\Delta), \phi_j(\Delta)\rangle = \dot{x}_j\langle \phi_j(\Delta), \phi_j(\Delta)\rangle - \sum_{i=0}^{N} x_i(t)\langle a(\Delta)\phi_i(\Delta), \phi_j(\Delta)\rangle$$

6. $\langle e(t,\Delta), \phi_j(\Delta)\rangle = 0$ implies

$$\dot{x}_j = \frac{1}{\langle \phi_j(\Delta), \phi_j(\Delta)\rangle}\sum_{i=0}^{N} x_i(t)\langle a(\Delta)\phi_i(\Delta), \phi_j(\Delta)\rangle$$

7. This gives use $N+1$ ordinary differential equations ($x \in \mathbb{R}$ in this example)

Example: First Order Linear System (contd.)

The equation

$$\dot{x}_j = \frac{1}{\langle \phi_j(\Delta), \phi_j(\Delta) \rangle} \sum_{i=0}^{N} x_i(t) \langle a(\Delta)\phi_i, \phi_j \rangle$$

in more compact form

$$\dot{x}_j = \frac{1}{\langle \phi_j(\Delta), \phi_j(\Delta) \rangle} \left[\langle a(\Delta)\phi_0(\Delta), \phi_j(\Delta) \rangle \quad \cdots \quad \langle a(\Delta)\phi_N(\Delta), \phi_j(\Delta) \rangle \right] \begin{pmatrix} x_0 \\ \vdots \\ x_N \end{pmatrix}$$

Define $\boldsymbol{x}_{pc} := (x_0 \; x_1 \; \cdots \; x_N)^T$, then

$$\dot{\boldsymbol{x}}_{pc} = \boldsymbol{A}_{pc} \boldsymbol{x}_{pc}$$

where

$$\boldsymbol{A}_{pc} := \boldsymbol{W}^{-1} \begin{bmatrix} \langle a(\Delta)\phi_0, \phi_0 \rangle & \cdots & \langle a(\Delta)\phi_N, \phi_0 \rangle \\ \vdots & & \vdots \\ \langle a(\Delta)\phi_0, \phi_N \rangle & \cdots & \langle a(\Delta)\phi_N, \phi_N \rangle \end{bmatrix}, \quad \boldsymbol{W} := \textbf{diag}\left(\langle \phi_0, \phi_0 \rangle \; \cdots \; \langle \phi_N, \phi_N \rangle \right)$$

Reduced Order System

Therefore

$$\underbrace{\dot{x} = -a(\Delta)x}_{\text{stochastic in } \mathbb{R}} \xrightarrow{\text{Polynomial Chaos}} \underbrace{\dot{\boldsymbol{x}}_{pc} = \boldsymbol{A}_{pc}\boldsymbol{x}_{pc}}_{\text{deterministic in } \mathbb{R}^{N+1}}$$

In general

$$\underbrace{\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\Delta})}_{\text{stochastic in } \mathbb{R}^n} \xrightarrow{\text{Polynomial Chaos}} \underbrace{\dot{\boldsymbol{x}}_{pc} = \boldsymbol{F}_{pc}(\boldsymbol{x}_{pc})}_{\text{deterministic in } \mathbb{R}^{n(N+1)}}$$

where $\boldsymbol{x}_{pc} := \begin{pmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_N \end{pmatrix}$ and $\hat{\boldsymbol{x}} = \sum_{i=0}^{N} \boldsymbol{x}_i(t)\phi_i(\Delta)$

Initial Condition Uncertainty

Transform uncertainty in dynamics as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\Delta}) \xrightarrow{\text{Polynomial Chaos}} \dot{\boldsymbol{x}}_{pc} = \boldsymbol{F}_{pc}(\boldsymbol{x}_{pc})$$

$$\boldsymbol{x}_{pc} := \begin{pmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_N \end{pmatrix} \text{ and } \hat{\boldsymbol{x}} = \sum_{i=0}^{N} \boldsymbol{x}_i(t)\phi_i(\boldsymbol{\Delta})$$

Let I.C. uncertainty be: $\boldsymbol{x}_0(\boldsymbol{\Delta})$

Initialize $\boldsymbol{x}_{pc}$ as

$$\boldsymbol{x}_i(t_0) := \langle \boldsymbol{x}_0(\boldsymbol{\Delta}), \phi_i(\boldsymbol{\Delta}) \rangle$$

Random variable $\boldsymbol{\Delta}$ is

$$\boldsymbol{\Delta} := \begin{pmatrix} \boldsymbol{\Delta}_0 \\ \boldsymbol{\Delta}_p \end{pmatrix}, \quad \begin{matrix} \boldsymbol{\Delta}_0 \text{ is I.C. uncertainty} \\ \boldsymbol{\Delta}_p \text{ is system parameter uncertainty} \end{matrix}$$

Basis functions $\phi_i(\boldsymbol{\Delta})$ are defined w.r.t $\boldsymbol{\Delta}$

Linear Systems

Consider Linear System

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{\Delta})\boldsymbol{x}, \text{ with } \boldsymbol{x}(t_0) := \boldsymbol{x}_0(\boldsymbol{\Delta}), \text{ and } \boldsymbol{\Delta} := \begin{pmatrix} \boldsymbol{\Delta}_0 \\ \boldsymbol{\Delta}_p \end{pmatrix}$$

- System has random parameters in $\boldsymbol{A}$ matrix and I.C.
- $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{\Delta} \in \mathbb{R}^d$
- Define basis function vector $\boldsymbol{\Phi}(\boldsymbol{\Delta}) := (\phi_0(\boldsymbol{\Delta}) \cdots \phi_N(\boldsymbol{\Delta}))^T$
- Approximate solution is

$$\hat{\boldsymbol{x}} := \sum_{i=0}^{N} \boldsymbol{x}_i \phi_i(\boldsymbol{\Delta}) = \boldsymbol{X}\boldsymbol{\Phi}(\boldsymbol{\Delta}),$$

$$\boldsymbol{X} = [\boldsymbol{x}_0 \; \boldsymbol{x}_1 \; \cdots \; \boldsymbol{x}_N] \in \mathbb{R}^{n \times (N+1)}$$

Linear Systems (contd.)

Approximate solution

$$\hat{\boldsymbol{x}} = \boldsymbol{X}\boldsymbol{\Phi}(\boldsymbol{\Delta}),\ \boldsymbol{X} = [\boldsymbol{x}_0\ \boldsymbol{x}_1\ \cdots\ \boldsymbol{x}_N]$$

Define

$$\boldsymbol{x}_{pc} := \textbf{vec}\left(\boldsymbol{X}\right) \equiv \begin{pmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_N \end{pmatrix}$$

Therefore,

$$\begin{aligned} \textbf{vec}\left(\hat{\boldsymbol{x}}\right) &= \textbf{vec}\left(\boldsymbol{X}\boldsymbol{\Phi}\right) \\ \hat{\boldsymbol{x}} &= (\boldsymbol{\Phi}^T \otimes \boldsymbol{I}_n)\boldsymbol{x}_{pc} \qquad \textbf{vec}\left(ABC\right) \equiv (C^T \otimes A)\textbf{vec}\left(B\right) \end{aligned}$$

Linear Systems (contd.)

Residue

$$\begin{aligned}
\boldsymbol{e}(t,\boldsymbol{\Delta}) &:= \dot{\hat{\boldsymbol{x}}} - \boldsymbol{A}(\boldsymbol{\Delta})\hat{\boldsymbol{x}} = \dot{\boldsymbol{X}}\boldsymbol{\Phi}(\boldsymbol{\Delta}) - \boldsymbol{A}(\boldsymbol{\Delta})\boldsymbol{X} \\
\textbf{vec}\,(e) = \boldsymbol{e} &= \textbf{vec}\left(\dot{\boldsymbol{X}}\boldsymbol{\Phi}(\boldsymbol{\Delta}) - \boldsymbol{A}(\boldsymbol{\Delta})\boldsymbol{X}\boldsymbol{\Phi}(\boldsymbol{\Delta})\right) \\
&= (\boldsymbol{\Phi}^T \otimes \boldsymbol{I}_n)\dot{\boldsymbol{x}}_{pc} - \left(\boldsymbol{\Phi}^T(\boldsymbol{\Delta}) \otimes \boldsymbol{A}(\boldsymbol{\Delta})\right)\boldsymbol{x}_{pc}
\end{aligned}$$

$\langle \boldsymbol{e}, \phi_i(\boldsymbol{\Delta})\rangle = 0$ implies

$$\dot{\boldsymbol{x}}_i = \left(\langle \phi_i(\boldsymbol{\Delta}), \phi_i(\boldsymbol{\Delta})\rangle \otimes \boldsymbol{I}_n\right)^{-1} \left\langle \boldsymbol{\Phi}^T(\boldsymbol{\Delta}) \otimes \boldsymbol{A}(\boldsymbol{\Delta}), \phi_i(\boldsymbol{\Delta})\right\rangle \boldsymbol{x}_{pc}$$

Linear Systems (contd.)

Deterministic linear dynamics

$$\dot{\boldsymbol{x}}_{pc} = \boldsymbol{A}_{pc}\,\boldsymbol{x}_{pc}$$

$\boldsymbol{x}_{pc} \in \mathbb{R}^{n(N+1)}$, $\boldsymbol{A}_{pc} \in \mathbb{R}^{n(N+1)\times n(N+1)}$

A_{pc} is defined as

$$\boldsymbol{A}_{pc} := (\boldsymbol{W} \otimes \boldsymbol{I}_n)^{-1} \begin{bmatrix} \left\langle \boldsymbol{\Phi}^T \otimes \boldsymbol{A}(\boldsymbol{\Delta}), \phi_0 \right\rangle \\ \vdots \\ \left\langle \boldsymbol{\Phi}^T \otimes \boldsymbol{A}(\boldsymbol{\Delta}), \phi_N \right\rangle \end{bmatrix}$$

Recall

$$\boldsymbol{W} := \textbf{diag}\left(\langle \phi_0, \phi_0 \rangle \ \cdots \ \langle \phi_N, \phi_N \rangle\right)$$

Computation of Mean

Given $\boldsymbol{x}(\boldsymbol{\Delta}) := \boldsymbol{X}\boldsymbol{\Phi}(\Delta)$

$$\begin{aligned}\mathbf{E}\left[\boldsymbol{x}(\boldsymbol{\Delta})\right] &= \mathbf{E}\left[\boldsymbol{X}\boldsymbol{\Phi}(\Delta)\right]\\ &= \boldsymbol{X}\mathbf{E}\left[\boldsymbol{\Phi}(\Delta)\right]\\ &= \boldsymbol{X}\left(1\ 0\ \cdots\ 0\right)^T\\ &= \boldsymbol{x}_0\end{aligned}$$

Also

$$\mathbf{E}\left[\boldsymbol{x}(\boldsymbol{\Delta})\right] = \mathbf{E}\left[(\boldsymbol{\Phi}^T \otimes \boldsymbol{I}_n)\boldsymbol{x}_{pc}\right] = \left(\mathbf{E}\left[\boldsymbol{\Phi}^T\right] \otimes \boldsymbol{I}_n\right)\boldsymbol{x}_{pc} = \left(\boldsymbol{F}^T \otimes \boldsymbol{I}_n\right)\boldsymbol{x}_{pc}$$

where $\boldsymbol{F}^T = (1\ 0\ \cdots\ 0)$.

Computation of Variance

Given $\boldsymbol{x}(\boldsymbol{\Delta}) := \boldsymbol{X}\boldsymbol{\Phi}(\Delta)$

$$
\begin{aligned}
\boldsymbol{x}(\boldsymbol{\Delta})\boldsymbol{x}^T(\boldsymbol{\Delta}) &= \boldsymbol{X}\boldsymbol{\Phi}(\Delta)\boldsymbol{\Phi}^T(\boldsymbol{\Delta})\boldsymbol{X}^T \\
\mathsf{E}\left[\boldsymbol{x}(\boldsymbol{\Delta})\boldsymbol{x}^T(\boldsymbol{\Delta})\right] &= \mathsf{E}\left[\boldsymbol{X}\boldsymbol{\Phi}(\Delta)\boldsymbol{\Phi}^T(\boldsymbol{\Delta})\boldsymbol{X}^T\right] \\
&= \boldsymbol{X}\mathsf{E}\left[\boldsymbol{\Phi}(\Delta)\boldsymbol{\Phi}^T(\boldsymbol{\Delta})\right]\boldsymbol{X}^T \\
&= \boldsymbol{X}\begin{bmatrix} \langle\phi_0,\phi_0\rangle & 0 & \dots & 0 \\ 0 & \langle\phi_1,\phi_1\rangle & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \langle\phi_N,\phi_N\rangle \end{bmatrix}\boldsymbol{X}^T \\
&= \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^T
\end{aligned}
$$

Then

$$
\mathbf{Var}\left[\boldsymbol{x}\right] := \mathsf{E}\left[\left(\boldsymbol{x}-\mathsf{E}\left[\boldsymbol{x}\right]\right)\left(\boldsymbol{x}-\mathsf{E}\left[\boldsymbol{x}\right]\right)^T\right] = \boldsymbol{X}(\boldsymbol{W}-\boldsymbol{F}\boldsymbol{F}^T)\boldsymbol{X}^T
$$

Computation of Statistics -- summary

Mean

$$\mathsf{E}[x] = \boldsymbol{X}\boldsymbol{F} = \boldsymbol{x}_0$$

Variance

$$\mathbf{Var}[x] = \boldsymbol{X}(\boldsymbol{W} - \boldsymbol{F}\boldsymbol{F}^T)\boldsymbol{X}^T$$

where

$$\boldsymbol{F} = \mathsf{E}[\boldsymbol{\Phi}(\boldsymbol{\Delta})] = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \boldsymbol{W} = \mathsf{E}\left[\boldsymbol{\Phi}\boldsymbol{\Phi}^T\right] = \begin{bmatrix} \langle \phi_0, \phi_0 \rangle & 0 & \dots & 0 \\ 0 & \langle \phi_1, \phi_1 \rangle & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \langle \phi_N, \phi_N \rangle \end{bmatrix}$$

Polynomial Nonlinearity

Polynomials $x^n(\boldsymbol{\Delta})$, $x \in \mathbb{R}$ can be written as

$$\begin{aligned} x^n(\boldsymbol{\Delta}) &= (\boldsymbol{X}\boldsymbol{\Phi}(\boldsymbol{\Delta}))^n \\ \langle x^n(\boldsymbol{\Delta}), \phi_i(\boldsymbol{\Delta}) \rangle &= \langle (\boldsymbol{X}\boldsymbol{\Phi}(\boldsymbol{\Delta}))^n, \phi_i \rangle \\ &= \sum_{i_1=0}^{N} \cdots \sum_{i_n=0}^{N} x_{i_1} \cdots x_{i_n} \langle \phi_{i_1} \cdots \phi_{i_n}, \phi_i \rangle \end{aligned}$$

- Essentially integration of polynomials
  - analytical or numerical (exact).
- Inner product $\langle \phi_{i_1} \cdots \phi_{i_n}, \phi_i \rangle$
  - can be computed offline
  - stored in sparse, symmetric tensor

Rational polynomials

Functions such as $\frac{x^n(\mathbf{\Delta})}{y^m(\mathbf{\Delta})}$, $x, y \in \mathbb{R}$ can be approximated as

$$
\begin{aligned}
z(\mathbf{\Delta}) &= \frac{x^n(\mathbf{\Delta})}{y^m(\mathbf{\Delta})} \\
\boldsymbol{Z}\boldsymbol{\Phi}(\mathbf{\Delta}) &= \frac{(\boldsymbol{X}\boldsymbol{\Phi}(\mathbf{\Delta}))^n}{(\boldsymbol{Y}\boldsymbol{\Phi}(\mathbf{\Delta}))^m} \\
(\boldsymbol{Y}\boldsymbol{\Phi})^m\, \boldsymbol{Z}\boldsymbol{\Phi} &= (\boldsymbol{X}\boldsymbol{\Phi})^n \\
\langle(\boldsymbol{Y}\boldsymbol{\Phi})^m\, \boldsymbol{Z}\boldsymbol{\Phi}, \phi_i)\rangle &= \langle(\boldsymbol{X}\boldsymbol{\Phi})^n, \phi_i\rangle, \quad i = \{0, 1, \cdots, N\}
\end{aligned}
$$

Given $\boldsymbol{X}, \boldsymbol{Y}$ solve system of linear equations to obtain $\boldsymbol{Z}$

$$
\begin{bmatrix} \left\langle \boldsymbol{\Phi}^T \otimes (\boldsymbol{Y}\boldsymbol{\Phi})^m, \phi_0 \right\rangle \\ \vdots \\ \left\langle \boldsymbol{\Phi}^T \otimes (\boldsymbol{Y}\boldsymbol{\Phi})^m, \phi_N \right\rangle \end{bmatrix} \boldsymbol{z}_{pc} = \begin{pmatrix} \langle(\boldsymbol{X}\boldsymbol{\Phi})^n, \phi_0\rangle \\ \vdots \\ \langle(\boldsymbol{X}\boldsymbol{\Phi})^m, \phi_N\rangle \end{pmatrix}
$$

Polynomial integrations

Transcendental Functions

Let $f(x)$ be a transcendental function:

- e.g. $x^a, e^x, x^{1/x}, \log(x), \sin(x)$, etc.

Use Taylor series expansion about mean

- Define $x := x_0 + d$, d is deviation from mean x_0
- Expand

$$f(x) = f(x_0 + d) = f(x_0) + f'(x_0)d + f''(x_0)\frac{d^2}{2!} + \cdots$$

- $x(\boldsymbol{\Delta}) := x_0 + \underbrace{\sum_{i=1}^{N} x_i \phi_i(\boldsymbol{\Delta})}_{d(\boldsymbol{\Delta})}$
- Therefore

$$\langle f(x(\boldsymbol{\Delta})), \phi_i(\boldsymbol{\Delta})\rangle \approx f(x_0)\langle 1, \phi_i\rangle + f'(x_0)\langle d, \phi_i\rangle + \frac{f''(x_0)}{2!}\langle d^2, \phi_i\rangle + \cdots$$

Transcendental Functions (contd.)

Taylor Series Approximation

$$\langle f(x(\boldsymbol{\Delta})), \phi_i(\boldsymbol{\Delta})\rangle \approx f(x_0)\langle 1, \phi_i\rangle + f'(x_0)\langle d, \phi_i\rangle + \frac{f''(x_0)}{2!}\langle d^2, \phi_i\rangle + \cdots$$

- $\langle d^n, \phi_i\rangle$ is integration of polynomials
- Straightforward
- Computationally efficient
- Severe inaccuracies for higher order PC approximations

Remedies

- Approximate $f(x)$ using polynomials, piecewise polynomials
- Non-intrusive: multi-dimensional integrals via sampling, tensor-product quadrature, Smolyak sparse grid, or cubature
- Regression Approach: $\mathcal{L}_2$ optimization

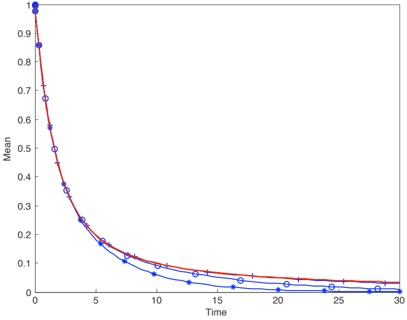
Example: First Order Linear System

Dynamics:

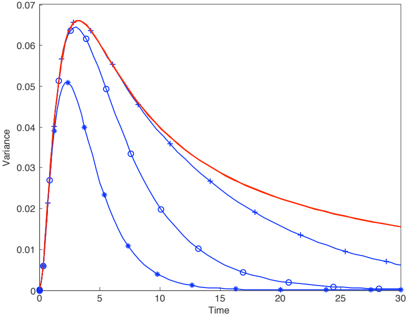
$$\dot{x} = -a(\Delta)x, \qquad a \in \mathcal{U}_{[0,1]}$$

Analytical Statistics:

$$\bar{x}(t) = \frac{1-e^{-t}}{t}, \qquad \sigma(t) = \frac{1-e^{-2t}}{2t} - \left(\frac{1-e^{-t}}{t}\right)^2$$



(a) Mean.



(b) Variance.

Errors Due to Finite Terms

Dynamics:

$$\dot{x} = -a(\Delta)x, \qquad a \in \mathcal{U}_{[0,1]}$$

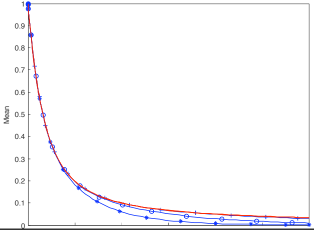
Analytical Solution:

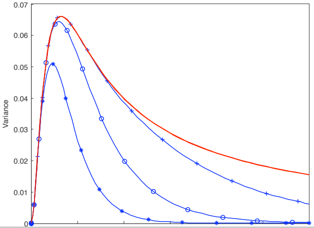
$$x(t,\Delta) = x(t_0)e^{-a(\Delta)t}$$

PC Solution:

$$\hat{x}(t,\Delta) = \sum_{i=0}^{P} x_i(t)\phi_i(\Delta)$$

Error: Finite term approximation of exponential.



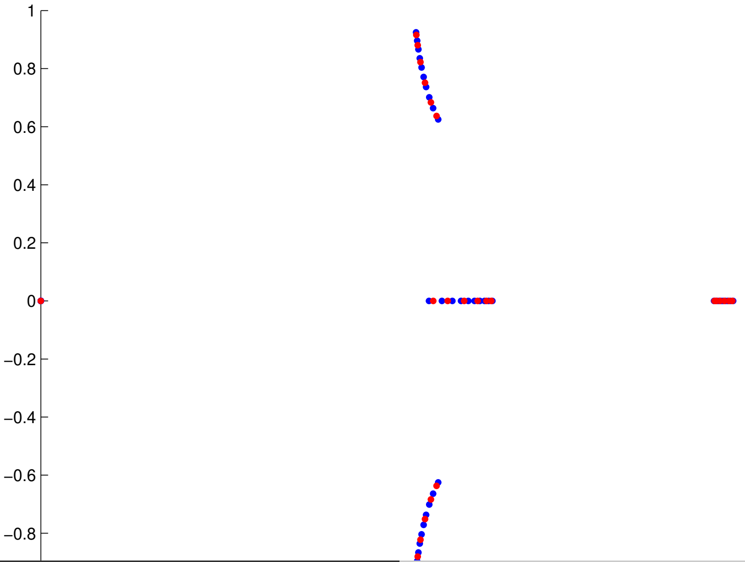


Example: Eigen Analysis -- Linear F-16 Aircraft

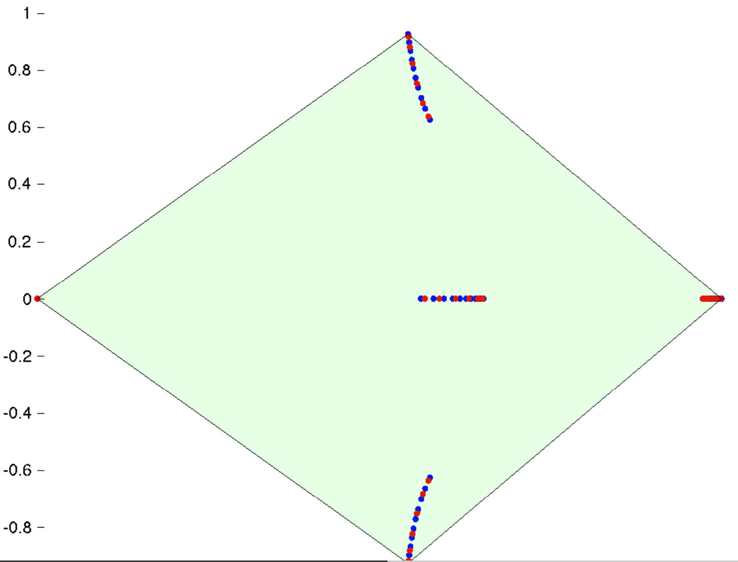
$$A(\boldsymbol{\Delta}) = \begin{bmatrix} 0.1658 & -13.1013 & -7.2748(1+0.2\Delta) & -32.1739 & 0.2780 \\ 0.0018 & -0.1301 & 0.9276(1+0.2\Delta) & 0 & -0.0012 \\ 0 & -0.6436 & -0.4763 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

- Linearized about flight condition $V = 160\ ft/s$ and $\alpha = 35^o$
- Uncertainty due to damping term C_{xq}
- Difficult to model at high angle of attack
- 20% uncertainty about nominal

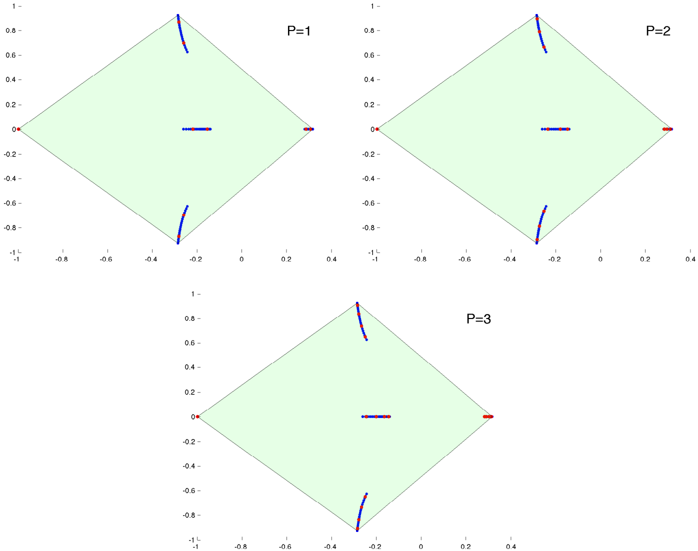
Example: Eigen Analysis -- Linear F-16 Aircraft



Example: Eigen Analysis -- Linear F-16 Aircraft



Example: Spread of Spectrum -- Linear F-16 Aircraft



Example: Nonlinear System -- Lorenz Attractor

Dynamics:

$$\dot{x} = \sigma(y - x), \qquad \dot{y} = x(\rho - z) - y, \qquad \dot{z} = xy - \beta z.$$

Initial Condition:

$$[x, y, z]^T = [1.50887, -1.531271, 25.46091]^T$$

Parameters:

$$\sigma = 10(1 + 0.1\Delta_1), \qquad \rho = 28(1 + 0.1\Delta_2), \qquad \beta = 8/3, \qquad \boldsymbol{\Delta} \in \mathcal{U}_{[-1,1]^2}.$$

$$x(t, \boldsymbol{\Delta}) \approx \sum_{i=0}^{P} x_i(t)\phi_i(\boldsymbol{\Delta}) \qquad \langle\phi_k^2\rangle\dot{x}_k(t) = \sum_{i=0}^{P}\langle\sigma\phi_i\phi_k\rangle(y_i - x_i)$$

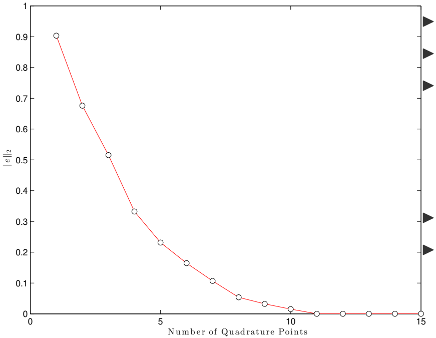
$$y(t, \boldsymbol{\Delta}) \approx \sum_{i=0}^{P} y_i(t)\phi_i(\boldsymbol{\Delta}) \qquad \langle\phi_k^2\rangle\dot{y}_k(t) = \sum_{i=0}^{P}\langle\rho\phi_i\phi_k\rangle x_i - \sum_{i=0}^{P}\sum_{j=0}^{P}\langle\phi_i\phi_j\phi_k\rangle x_i z_j - \langle\phi_k^2\rangle y_k$$

$$z(t, \boldsymbol{\Delta}) \approx \sum_{i=0}^{P} z_i(t)\phi_i(\boldsymbol{\Delta}) \qquad \langle\phi_k^2\rangle\dot{z}_k(t) = \sum_{i=0}^{P}\sum_{j=0}^{P}\langle\phi_i\phi_j\phi_k\rangle x_i y_j - \beta\langle\phi_k^2\rangle z_k$$

Example: Nonlinear System -- Lorenz Attractor

Integrals:

$$\langle \phi_k(\boldsymbol{\Delta})^2 \rangle$$
$$\langle \sigma(\boldsymbol{\Delta})\phi_i(\boldsymbol{\Delta})\phi_k(\boldsymbol{\Delta}) \rangle$$
$$\langle \rho(\boldsymbol{\Delta})\phi_i(\boldsymbol{\Delta})\phi_k(\boldsymbol{\Delta}) \rangle$$
$$\langle \phi_i(\boldsymbol{\Delta})\phi_j(\boldsymbol{\Delta})\phi_k(\boldsymbol{\Delta}) \rangle$$



- analytical
- numerical
  - non intrusive (blackbox)
  - quadratures defined by roots of $\phi_N(\cdot)$
  - tensor product of univariate quadratures
  - Here we use 7^{th} order PC approximation
  - Highest order polynomial integrated is 21 in $\langle \phi_i(\boldsymbol{\Delta})\phi_j(\boldsymbol{\Delta})\phi_k(\boldsymbol{\Delta}) \rangle$
  - $N = 11$ will exactly integrate polynomials of order ≤ 22,i.e.

$$\langle \phi_i(\boldsymbol{\Delta})\phi_j(\boldsymbol{\Delta})\phi_k(\boldsymbol{\Delta}) \rangle = \sum_r w_r \phi_i(\boldsymbol{\Delta}_r)\phi_j(\boldsymbol{\Delta}_r)\phi_k(\boldsymbol{\Delta}_r)$$

  - Approximate for non polynomial integrands
  - Multidimensional moments can be computed efficiently from products of one dimensional moments

    multivariate ϕ_i's are tensor products of univariate functions

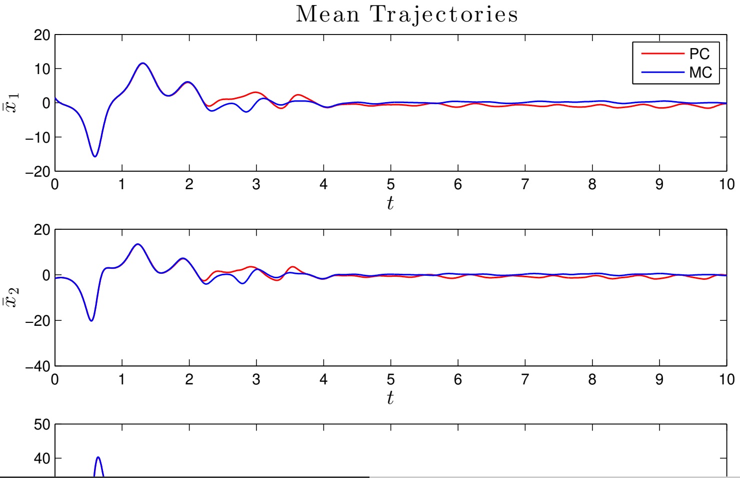
Example: Nonlinear System -- Lorenz Attractor

MC: 1000 samples

- using MATLAB `rand(...)`

PC: 7^{th} order approximation

- 36 basis functions



Stochastic Collocation

Raktim Bhattacharya

Aerospace Engineering, Texas A&M University

isrlab.github.io

Basic Idea

- Sample domain $\mathcal{D}_{\mathbf{\Delta}}$ suitably
  - roots of basis functions – $\phi(\mathbf{\Delta})$ same as Galerkin projection
  - multi-dimension samples $\Leftrightarrow$ tensor product of roots or sparse grid
- Enforce stochastic dynamics at each sample point
  - Time varying coefficient at each sample point
- Interpolate (Lagrangian) for intermediate points

Algorithm

1. Given stochastic dynamics with uncertainty $\boldsymbol{\Delta}$

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{\Delta})$$

2. For p^{th} order approximation:
   - sample domain $\mathcal{D}_{\boldsymbol{\Delta}}$ with roots of $p+1$ order polynomial
   - tensor grid, sparse grid, etc.
   - samples $\boldsymbol{\Delta} := \{\boldsymbol{\Delta}_i\}$, $i = 0, \cdots, p$.
3. Coefficient $\boldsymbol{x}_i$ evolves according to

$$\dot{\boldsymbol{x}}_i = f(\boldsymbol{x}_i, \boldsymbol{\Delta}_i), \text{ deterministic solution}$$

4. Approximate stochastic solution

$$\hat{\boldsymbol{x}}(t, \boldsymbol{\Delta}) := \sum_{i=0}^{p} \boldsymbol{x}_i(t) L_i(\boldsymbol{\Delta})$$

L_i are Lagrangian interpolants $L_i(y) = \prod_{j=0, j\neq i}^{p} \frac{y - y_j}{y_i - y_j}$.

Computation of Statistics

- Mean

$$\mathbf{E}\left[\boldsymbol{x}(t)\right] \approx \mathbf{E}\left[\sum_{i=0}^{p} \boldsymbol{x}_i(t) L_i(\boldsymbol{\Delta})\right] = \sum_{i=0}^{p} \boldsymbol{x}_i(t) \mathbf{E}\left[L_i(\boldsymbol{\Delta})\right]$$

- Computation of $\mathbf{E}\left[L_i(\boldsymbol{\Delta})\right]$ involves high-dimensional polynomial integration
  - analytical
  - numerical: quadratures, sparse grids, etc
- Higher order statistics: similar to computation of mean.

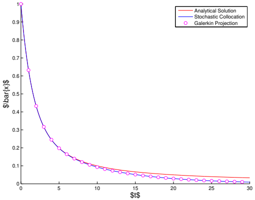
Example: Linear First Order System

Dynamics:

$$\dot{x} = -a(\Delta)x, \qquad a \in \mathcal{U}_{[0,1]}$$

Analytical Statistics:

$$\bar{x}(t) = \frac{1 - e^{-t}}{t},$$



(a) MC (3 samples)

Example: Nonlinear System -- Lorenz Attractor

Dynamics:

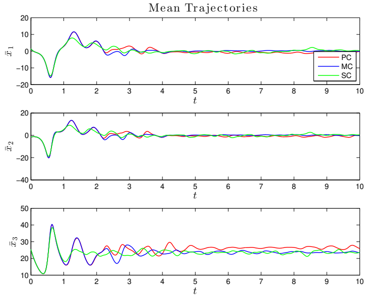
$$\dot{x} = \sigma(y - x), \qquad \dot{y} = x(\rho - z) - y, \qquad \dot{z} = xy - \beta z.$$

Initial Condition:

$$[x, y, z]^T = [1.50887, -1.531271, 25.46091]^T$$

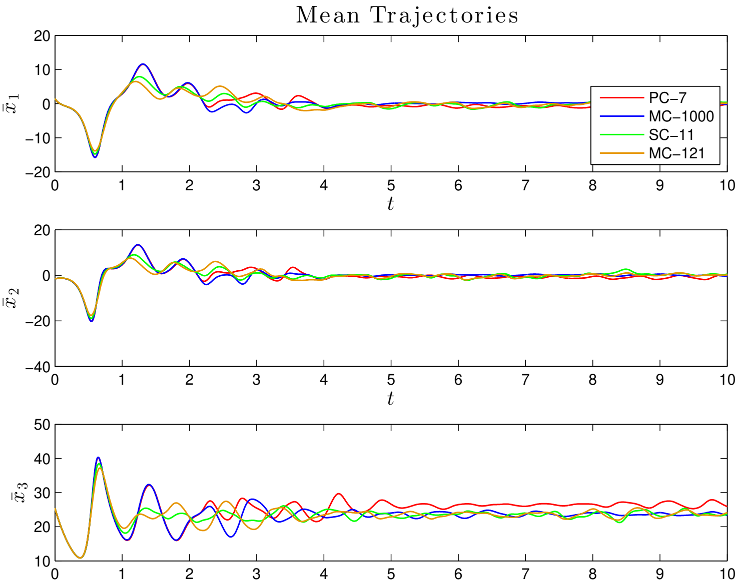
Parameters:

$$\sigma = 10(1 + 0.1\Delta_1), \qquad \rho = 28(1 + 0.1\Delta_2), \qquad \beta = 8/3, \qquad \boldsymbol{\Delta} \in \mathcal{U}_{[-1,1]^2}.$$



- MC: 1000 samples
  - using MATLAB `rand(...)`
- SC: 11 quadrature points
  - same as 7^{th} order PC
  - 121 grid points in 2D
- SC performance is poor for nonlinear systems!

Example: Nonlinear System -- Lorenz Attractor



Karhunen-Loève Expansion

Raktim Bhattacharya
Aerospace Engineering, Texas A&M University
isrlab.github.io

Basic Idea

Given a random process $X(t,\omega) := \{X_t(\omega)\}_{t\in[t_1,t_2]}$

- $X_t(\omega) \in \mathcal{L}_2(\Omega,\mathcal{F},P)$ finite second moment
  - $\mathcal{L}_2(\Omega,\mathcal{F},P) := \{X : \Omega \mapsto \mathbb{R} : \int_\Omega |X(\omega)|^2 dP(\omega) < \infty\}$
- Auto Correlation

$$R_X(t_1,t_2) := \mathbf{E}\left[X_{t_1} X_{t_2}\right]$$

- Auto Covariance

$$\begin{aligned} C_X(t_1,t_2) &:= R_X(t_1,t_2) - \mu_{t_1}\mu_{t_2} \\ &= R_X(t_1,t_2) - \mu^2 \quad \text{stationary} \end{aligned}$$

- $C_X(t_1,t_2)$ is bounded, symmetric and positive definite, thus

$$C_X(t_1,t_2) = \sum_{i=0}^{\infty} \lambda_i f_i(t_1) f_i(t_2) \quad \text{spectral decomposition}$$

where λ_i and $f_i(\cdot)$ are eigenvalues and eigenvectors of the covariance kernel.

Eigenvalues and Eigenfunctions

- λ_i and $f_i(\cdot)$ are solutions of

$$\int_{\mathcal{D}} C_X(t_1,t_2)\, f_i(t)\, dt_1 = \lambda_i\, f_i(t_2),$$ Fredholm integral equation of second kind

with $\displaystyle\int_{\mathcal{D}} f_i(t) f_j(t) dt = \delta_{ij}.$

- Write $X(t,\omega) := \bar{X}(t) + Y(t,\omega)$, where

$$Y(t,\omega) \overset{m.s.}{=} \sum_{i=0}^{\infty} \xi_i(\omega)\sqrt{\lambda_i}\, f_i(t), \text{ and } \xi_i(\omega) = \frac{1}{\lambda_i}\int_{\mathcal{D}} Y(t,\omega) f_i(t) dt.$$

- Reproducing Kernel Hilbert Space
  - Congruence between two Hilbert spaces!
  - $\{f_i(t)\} \mapsto X(t,\omega)$ or equivalently
  - $\{f_i(t)\} \mapsto \{\xi_i(\omega)\}$

Solution of Integral Equation

- Homogeneous Fredholm integral equation of the second kind,

$$\int_{\mathcal{D}} C_X(t_1, t_2)\, f_i(t)\, dt_1 = \lambda_i\, f_i(t_2) \quad \text{well studied problem}$$

- $C_X(t_1, t_2)$ is bounded, symmetric, and positive definite, implies
  1. The set $f_i(t)$ of eigenfunctions is orthogonal and complete.
  2. For each eigenvalue λ_k, there correspond at most a finite number of linearly independent eigenfunctions.
  3. There are at most a countably infinite set of eigenvalues.
  4. The eigenvalues are all positive real numbers.
  5. The kernel $C_X(t_1, t_2)$ admits of the following uniformly convergent expansion

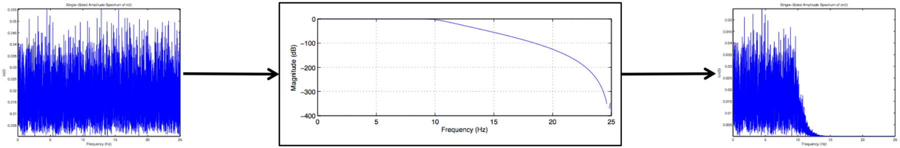
$$C_X(t_1, t_2) = \sum_{i=0}^{\infty} \lambda_i f_i(t_1) f_i(t_2)$$

- Applicable to wide range of processes

Rational Spectra: Special Case

- 1D random process
- Stationary output of a linear filter, excited by white noise

Filter 10Hz



White noise

Filtered noise

- Spectral density of the form $S(s^2) = H(j\omega)H(-j\omega) = \frac{N(s^2)}{D(s^2)}$

N and D are polynomials in s^2 such that

$$\int_{-\infty}^{\infty} S(-\omega^2)d\omega < \infty$$

$s = j\omega$, here ω is frequency

- Degree of $D(s^2)$ must exceed degree of $N(s^2)$ by at least two.
- No roots of $D(s^2)$ on the imaginary axis
- $S(\omega) \geq 0$, $\Rightarrow$ purely imaginary zeros of $N(s^2)$ of even multiplicity

- Finite dimensional Markovian process

Important Kernel

Study specific kernel

$$C_X(t_1, t_2) = e^{-c|t_1 - t_2|}$$

$1/c$ is the correlation time or length.

- Many applications.
- Other kernels also possible

Solve integral equation

$$\int_{\mathcal{D}} C_X(t_1, t_2)\, f_i(t)\, dt_1 = \lambda_i\, f_i(t_2).$$

Or equivalently solve

ODE: $\ddot{f}(t) + \omega^2 f(t) = 0,\ \ \omega^2 = \dfrac{2c - c^2\lambda}{\lambda},\ \ -a \le t \le a$

Boundary Condition: $cf(a) + \dot{f}(a) = 0,\ \ cf(-a) - \dot{f}(-a) = 0.$

Basis Functions

Equivalently solve for ω, ω^*

Odd i

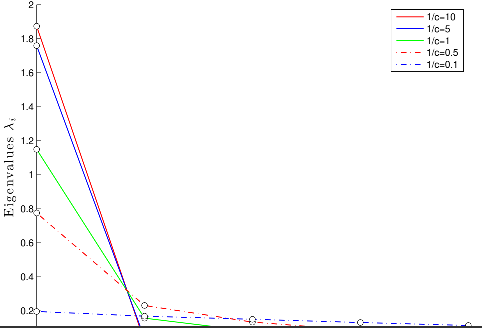
$$c - \omega \tan(\omega a) = 0, \quad \lambda_i = \frac{2c}{\omega_i^2 + c^2}$$

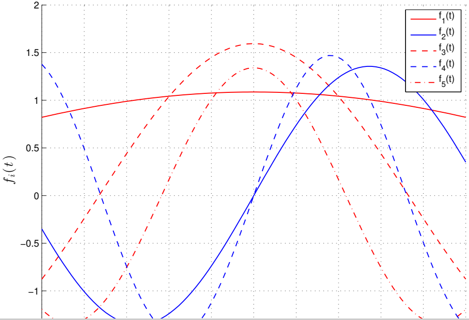
$$f_i(t) = \frac{\cos(\omega_i t)}{\sqrt{a + \frac{\sin(2\omega_i a)}{2\omega_i}}}$$

Even i

$$\omega^* + c \tan(\omega^* a) = 0, \quad \lambda_i^* = \frac{2c}{{\omega_i^*}^2 + c^2}$$

$$f_i^*(t) = \frac{\sin(\omega_i^* t)}{\sqrt{a - \frac{\sin(2\omega_i^* a)}{2\omega_i^*}}}$$





Coefficients

Recall $X(t,\omega) := \bar{X}(t) + Y(t,\omega)$, ω here is an event in the probability space $(\Omega, \mathcal{F}, P)$

$$
\begin{aligned}
Y(t,\omega) &\overset{m.s.}{=} \sum_{i=0}^{\infty} \xi_i(\omega)\sqrt{\lambda_i}\, f_i(t) \\
&= \sum_{i=0}^{\infty} \left[\xi_i(\omega)\sqrt{\lambda_i}\, f_i(t) + \xi_i^*(\omega)\sqrt{\lambda_i^*}\, f_i^*(t)\right]
\end{aligned}
$$

- $\xi_i(\omega), \xi_i^*(\omega)$ are uncorrelated random variables determined from $Y(t,\omega)$
- $\xi_i(\omega), \xi_i^*(\omega)$ model the distribution of amplitude of $Y(t,\omega)$
- $f_i(t), f_i^*(t)$ models the distribution of signal power over time or among frequencies

If $Y(t,\omega)$ is a Gaussian process

- $\xi_i(\omega), \xi_i^*(\omega)$ Gaussian independent random variables
- KL – expansion is almost surely convergent

UQ Application

Dynamical system with process noise $n(t,\omega)$

$$\dot{x} = f(t, \mathbf{\Delta}, x) + n(t,\omega)$$

Replace

$$n(t,\omega) \approx \sum_{i=0}^{N} \left[\xi_i(\omega)\sqrt{\lambda_i}\, f_i(t) + \xi_i^*(\omega)\sqrt{\lambda_i^*}\, f_i^*(t)\right]$$

Define new parameter vector

$$\mathbf{\Delta}' := \left(\mathbf{\Delta}^T, \xi_0, \xi_0^*, \cdots, \xi_N, \xi_N^*\right)^T$$

Rewrite dynamics as

$$\dot{x} = F(t, \mathbf{\Delta}', x),$$

Process noise converted to parametric uncertainty.

- Use PC, SC, or simplified FPK equation to determine $x(t, \mathbf{\Delta}')$
- Increases number of parameters $\Rightarrow$ increases computational complexity

Publications

1. J. Fisher, R. Bhattacharya, *Stability Analysis of Stochastic Systems using Polynomial Chaos*, American Control Conference 2008.
2. A. Prabhakar and R. Bhattacharya, *Analysis of Hypersonic Flight Dynamics with Probabilistic Uncertainty in System Parameters*, AIAA GNC 2008.
3. A. Prabhakar, J. Fisher, R. Bhattacharya, *Polynomial Chaos Based Analysis of Probabilistic Uncertainty in Hypersonic Flight Dynamics*, AIAA Journal of Guidance, Control, and Dynamics, Vol.33 No.1 (222-234), 2010.
4. J. Fisher, R. Bhattacharya, *Optimal Trajectory Generation with Probabilistic System Uncertainty Using Polynomial Chaos*, Journal of Dynamic Systems, Measurement and Control, volume 133, Issue 1, January 2011.
5. J. Fisher, R. Bhattacharya, *Linear Quadratic Regulation of Systems with Stochastic Parameter Uncertainties*, Automatica, 2009.
6. Roger G. Ghanem, Pol D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Revised Edition (Dover Civil and Mechanical Engineering
7. Olivier Le Maitre, Omar M Knio, *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*, Scientific Computation.
8. Dongbin Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, ISBN: 9780691142128, Princeton Press.