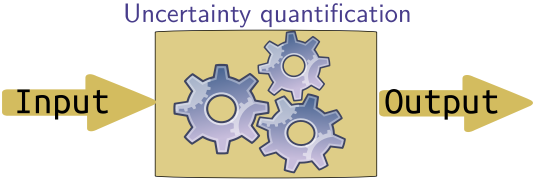
Polynomial chaos expansions for time-dependent problems

Chu V. Mai

Chair of Risk, Safety & Uncertainty Quantification

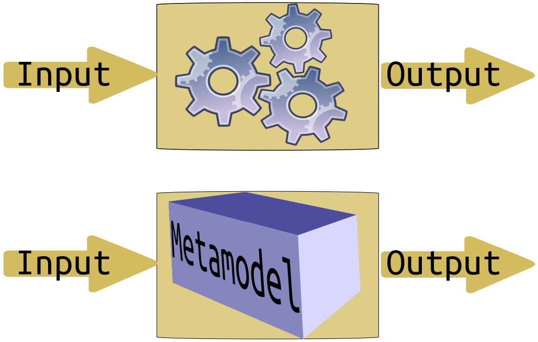
MascotNum, April 8th, 2015

Uncertainty quantification

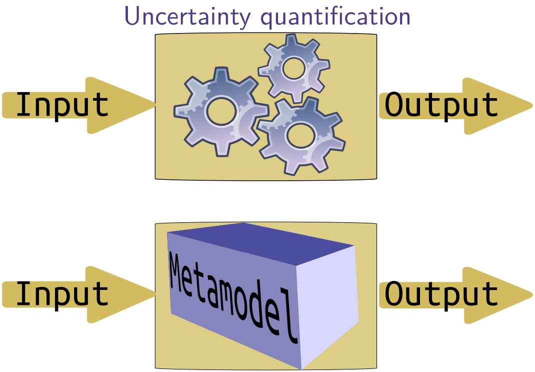


Uncertainty quantification

Uncertainty quantification



Uncertainty quantification



Polynomial Chaos Expansions

Outline

Outline

Polynomial chaos expansions

Ghanem and Spanos, 2003; Soize and Ghanem, 2004

PCE

$$Y(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} y_{\boldsymbol{\alpha}} \boldsymbol{\psi}_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$$

where:

- $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_M\}$ is the vector of uncertain input parameters,
- $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ is a multi-index,
- $\boldsymbol{\psi}_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ is a multivariate polynomial function

$$\boldsymbol{\psi}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \prod_{i=1}^{M} \psi_{\alpha_i}^{(i)}(\xi_i)$$

- $y_{\boldsymbol{\alpha}}$'s are expansion coefficients.

Polynomial chaos expansions

PCE for time-dependent problems

$$Y(t, \boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} y_{\boldsymbol{\alpha}}(t) \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$$

Frozen-in-time PCE

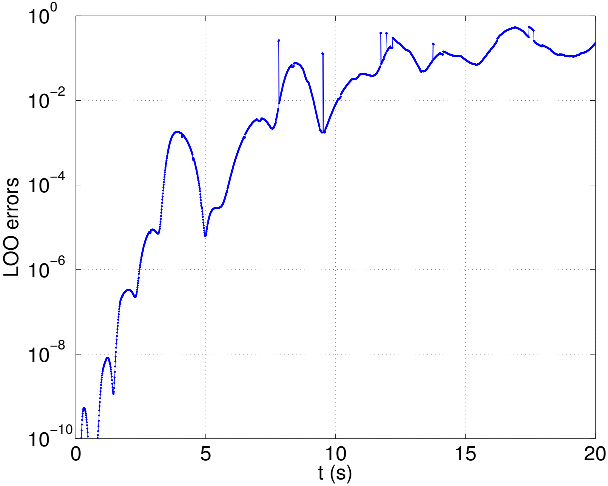
Computing PCE

- Intrusive approach — Le Maître and Knio, 2010
- Non-intrusive approach
  - Projection method
  - Least squares method

Leave-one-out (LOO) cross-validation

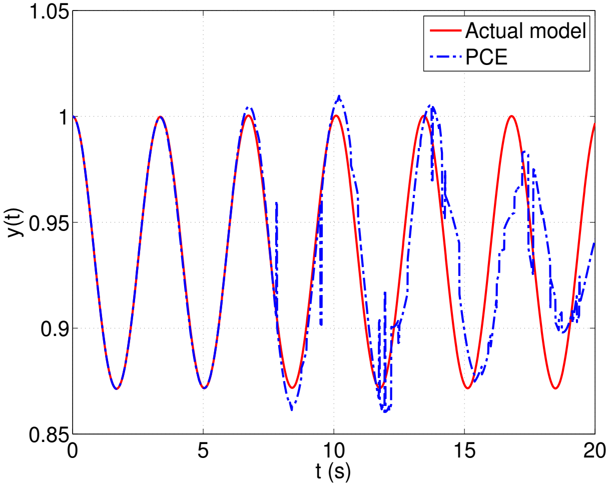
Adaptive sparse PCE based on least-angle-regression — Blatman and Sudret, 2011

PCE for time-dependent systems



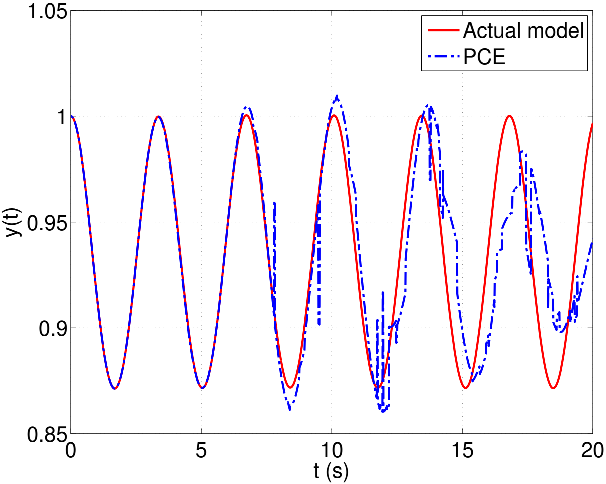
LOO errors *vs.* time (Rigid body dynamics)

PCE for time-dependent systems



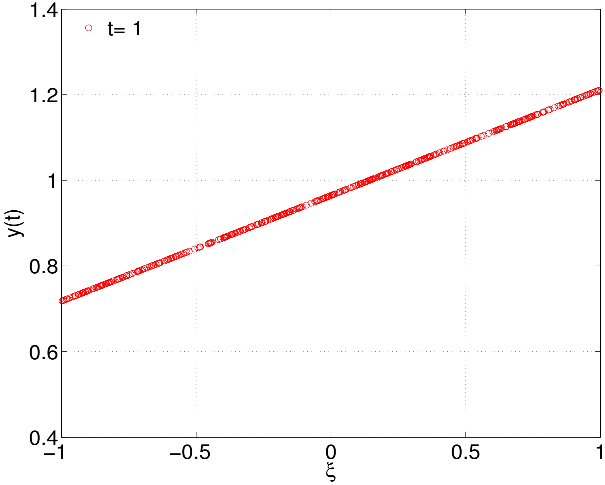
Prediction by PCE *vs.* actual response

PCE for time-dependent systems



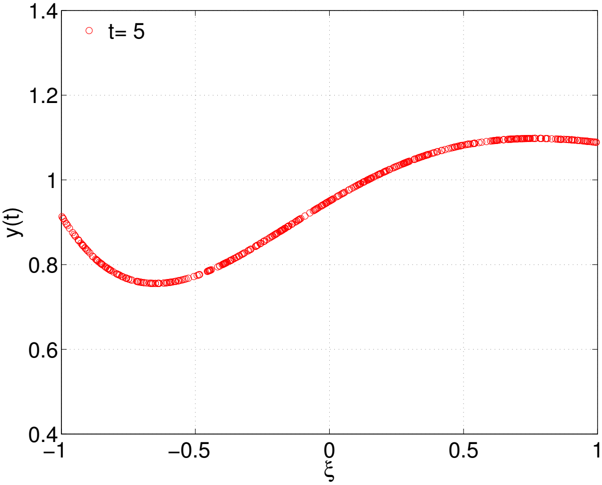
Problem: accuracy decreases over time

PCE for time-dependent systems



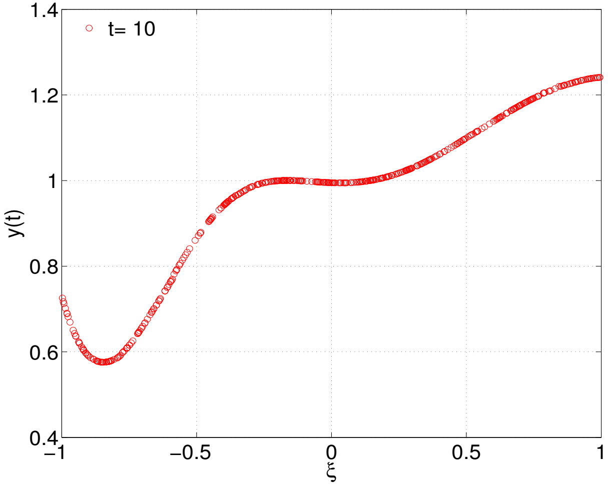
Time-dependent input-output relationship (Rigid body dynamics)

PCE for time-dependent systems



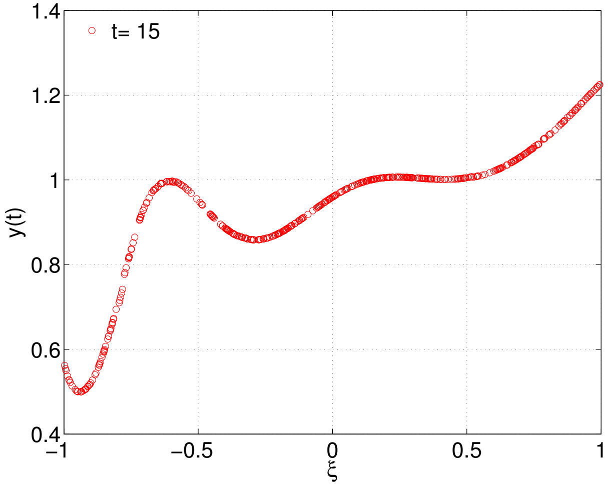
Time-dependent input-output relationship (Rigid body dynamics)

PCE for time-dependent systems



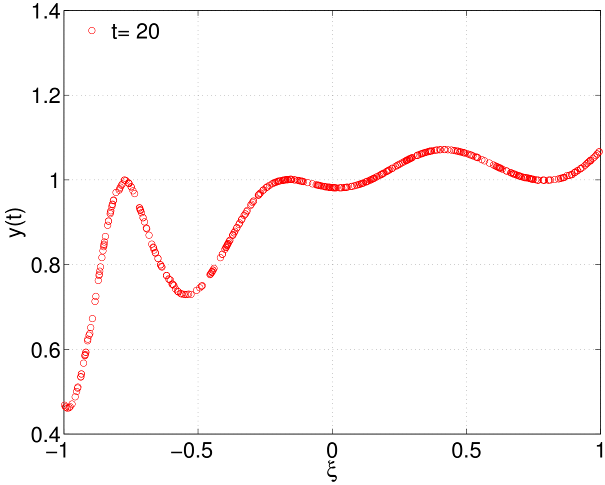
Time-dependent input-output relationship (Rigid body dynamics)

PCE for time-dependent systems



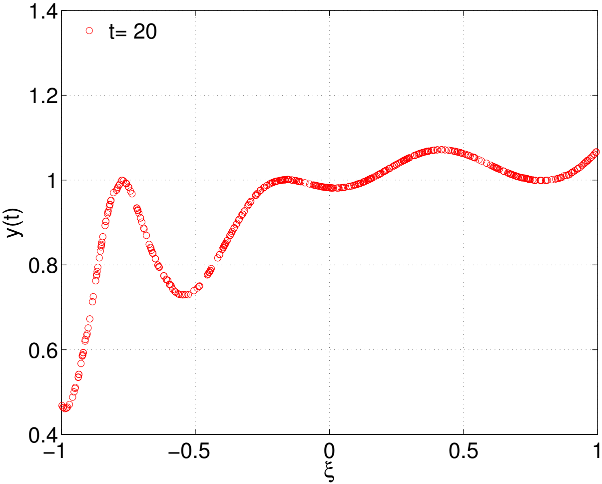
Time-dependent input-output relationship (Rigid body dynamics)

PCE for time-dependent systems



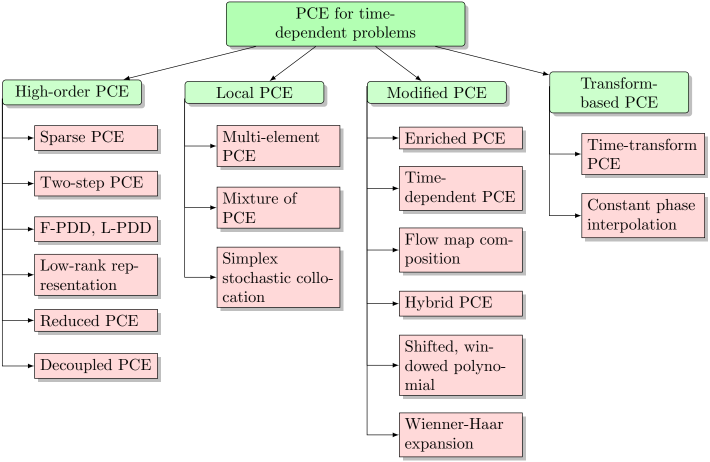
Time-dependent input-output relationship (Rigid body dynamics)

PCE for time-dependent systems



Complexity increases over time (nonlinearity, discontinuity, *etc.*)

PCE for time-dependent systems



Existing PCE approaches that can be used for time-dependent problems

Intrusive time-transform

Intrusive time-transform

Le Maître, Mathelin, et al., 2010

$$
\begin{aligned}
\frac{d\boldsymbol{x}^r}{dt}(t) &= \boldsymbol{f}^r(\boldsymbol{x}^r(t)),\\
\frac{d\boldsymbol{y}}{dt}(t;\boldsymbol{\xi}) &= \dot{\tau}(t;\boldsymbol{\xi})\boldsymbol{f}(\boldsymbol{y}(t;\boldsymbol{\xi});q(\boldsymbol{\xi})),\\
\frac{d\dot{\tau}}{dt}(t;\boldsymbol{\xi}) &= -\alpha_0\dot{\tau}(t;\boldsymbol{\xi})\Delta(t;\boldsymbol{\xi}) + \alpha_1\left[1-\dot{\tau}(t;\boldsymbol{\xi})\right],\\
\frac{d\tau}{dt}(t;\boldsymbol{\xi}) &= \dot{\tau}(t;\boldsymbol{\xi}),
\end{aligned}
$$

Application: limit-cycle oscillations

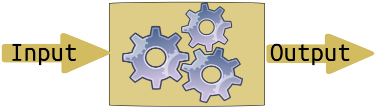
Intrusive time-transform

Intrusive time-transform

Le Maître, Mathelin, et al., 2010

$$\begin{aligned}
\frac{d\boldsymbol{x}^r}{dt}(t) &= \boldsymbol{f}^r(\boldsymbol{x}^r(t)), \\
\frac{d\boldsymbol{y}}{dt}(t;\boldsymbol{\xi}) &= \dot{\tau}(t;\boldsymbol{\xi})\boldsymbol{f}(\boldsymbol{y}(t;\boldsymbol{\xi});q(\boldsymbol{\xi})), \\
\frac{d\dot{\tau}}{dt}(t;\boldsymbol{\xi}) &= -\alpha_0\dot{\tau}(t;\boldsymbol{\xi})\Delta(t;\boldsymbol{\xi}) + \alpha_1\left[1-\dot{\tau}(t;\boldsymbol{\xi})\right], \\
\frac{d\tau}{dt}(t;\boldsymbol{\xi}) &= \dot{\tau}(t;\boldsymbol{\xi}),
\end{aligned}$$

Application: limit-cycle oscillations



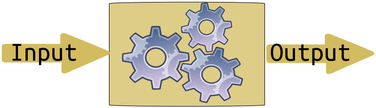
Intrusive time-transform

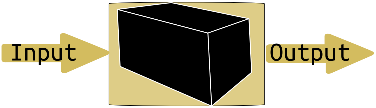
Intrusive time-transform

Le Maître, Mathelin, et al., 2010

$$\begin{aligned}
\frac{d\boldsymbol{x}^r}{dt}(t) &= \boldsymbol{f}^r(\boldsymbol{x}^r(t)),\\
\frac{d\boldsymbol{y}}{dt}(t;\boldsymbol{\xi}) &= \dot{\tau}(t;\boldsymbol{\xi})\boldsymbol{f}(\boldsymbol{y}(t;\boldsymbol{\xi});q(\boldsymbol{\xi})),\\
\frac{d\dot{\tau}}{dt}(t;\boldsymbol{\xi}) &= -\alpha_0\dot{\tau}(t;\boldsymbol{\xi})\Delta(t;\boldsymbol{\xi}) + \alpha_1\left[1-\dot{\tau}(t;\boldsymbol{\xi})\right],\\
\frac{d\tau}{dt}(t;\boldsymbol{\xi}) &= \dot{\tau}(t;\boldsymbol{\xi}),
\end{aligned}$$

Application: limit-cycle oscillations





Outline

1. PCE for time-dependent systems
2. Non-intrusive stochastic time transform
3. Numerical examples
4. Conclusions and perspective

Stochastic time transform

Focus on

- Oscillatory systems: limit-cycle oscillators, harmonic oscillators, *etc.*

System of type "black box"

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi}, t)$$

in which

- $\boldsymbol{x}(t=0) = \boldsymbol{x}_0$: initial condition
- $\boldsymbol{\xi}$: parameter vector of independent second-order random variables defined in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$

What to to

- Represent the system's output in a transformed time scale in which the similarity in the frequency and phase content of distinct trajectories is maximized.

Similarity measure

Similarity measure between two trajectories $x_1(t)$ and $x_2(t)$:

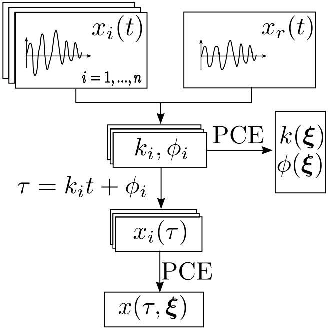
$$f(x_1(t), x_2(t)) = \frac{\left| \int_0^T x_1(t) x_2(t) \mathrm{d}t \right|}{\|x_1(t)\| \|x_2(t)\|}$$

- $\int_0^T x_1(t) x_2(t) \mathrm{d}t$ is the inner product of the two considered time histories
- $\| \cdot \|$ is the associated l^2-norm

Stochastic time transform

Time-transform PCE

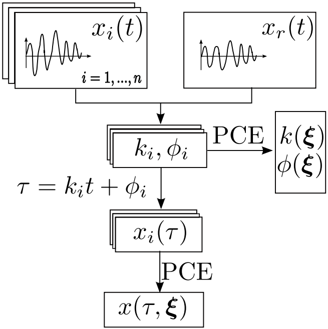
- Choose a reference trajectory $x_r(t)$
- Define a linear stochastic time transform $\tau = k\,t + \phi$
- For $i = 1, \ldots, n$,
  - Determine $\{k_i, \phi_i\}$ as solution of an optimization problem.
  - Represent $x_i(t)$ on the transformed time line τ, yielding $x_i(\tau)$.
- Compute PCE $x(\tau, \boldsymbol{\xi}) = \sum x_{\boldsymbol{\alpha}}(\tau)\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ using $\{x_i(\tau, \boldsymbol{\xi}), i = 1, \ldots, n\}$
- Compute PCE $k(\boldsymbol{\xi}) = \sum k_{\boldsymbol{\beta}}\psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi}) = \sum \varphi_{\boldsymbol{\gamma}}\psi_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ using $\{k_i, \phi_i, i = 1, \ldots, n\}$



Stochastic time transform

Time-transform PCE

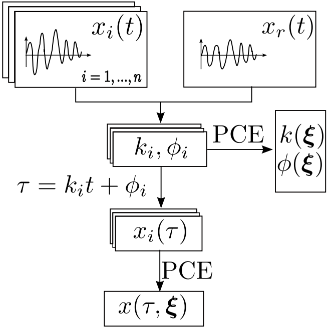
- Choose a reference trajectory $x_r(t)$
- Define a linear stochastic time transform $\tau = k\,t + \phi$
- For $i = 1, \ldots, n$,
  - Determine $\{k_i, \phi_i\}$ as solution of an optimization problem.
  - Represent $x_i(t)$ on the transformed time line τ, yielding $x_i(\tau)$.
- Compute PCE $x(\tau, \boldsymbol{\xi}) = \sum x_{\boldsymbol{\alpha}}(\tau)\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ using $\{x_i(\tau, \boldsymbol{\xi}), i = 1, \ldots, n\}$
- Compute PCE $k(\boldsymbol{\xi}) = \sum k_{\boldsymbol{\beta}}\psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi}) = \sum \varphi_{\boldsymbol{\gamma}}\psi_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ using $\{k_i, \phi_i, i = 1, \ldots, n\}$



Stochastic time transform

Time-transform PCE

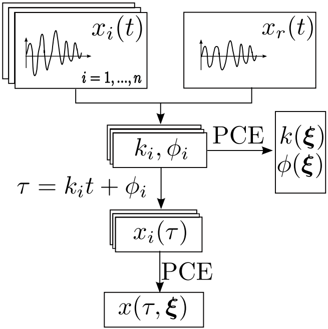
- Choose a reference trajectory $x_r(t)$
- Define a linear stochastic time transform $\tau = k\,t + \phi$
- For $i = 1, \ldots, n$,
  - Determine $\{k_i, \phi_i\}$ as solution of an optimization problem.
  - Represent $x_i(t)$ on the transformed time line τ, yielding $x_i(\tau)$.
- Compute PCE $x(\tau, \boldsymbol{\xi}) = \sum x_{\boldsymbol{\alpha}}(\tau)\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ using $\{x_i(\tau, \boldsymbol{\xi}), i = 1, \ldots, n\}$
- Compute PCE $k(\boldsymbol{\xi}) = \sum k_{\boldsymbol{\beta}}\psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi}) = \sum \varphi_{\boldsymbol{\gamma}}\psi_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ using $\{k_i, \phi_i, i = 1, \ldots, n\}$



Stochastic time transform

Time-transform PCE

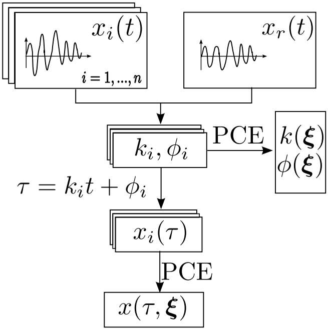
- Choose a reference trajectory $x_r(t)$
- Define a linear stochastic time transform $\tau = k\,t + \phi$
- For $i = 1, \ldots, n$,
  - Determine $\{k_i, \phi_i\}$ as solution of an optimization problem.
  - Represent $x_i(t)$ on the transformed time line τ, yielding $x_i(\tau)$.
- Compute PCE $x(\tau, \boldsymbol{\xi}) = \sum x_\alpha(\tau)\psi_\alpha(\boldsymbol{\xi})$ using $\{x_i(\tau, \boldsymbol{\xi}), i = 1, \ldots, n\}$
- Compute PCE $k(\boldsymbol{\xi}) = \sum k_\beta \psi_\beta(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi}) = \sum \varphi_\gamma \psi_\gamma(\boldsymbol{\xi})$ using $\{k_i, \phi_i, i = 1, \ldots, n\}$



Stochastic time transform

Time-transform PCE

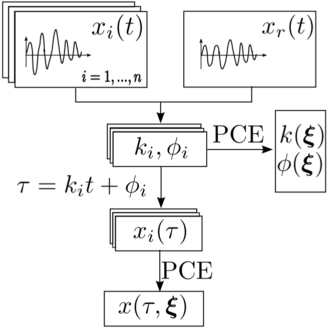
- Choose a reference trajectory $x_r(t)$
- Define a linear stochastic time transform $\tau = k\,t + \phi$
- For $i = 1, \ldots, n$,
  - Determine $\{k_i, \phi_i\}$ as solution of an optimization problem.
  - Represent $x_i(t)$ on the transformed time line τ, yielding $x_i(\tau)$.
- Compute PCE $x(\tau, \boldsymbol{\xi}) = \sum x_\alpha(\tau)\psi_\alpha(\boldsymbol{\xi})$ using $\{x_i(\tau, \boldsymbol{\xi}), i = 1, \ldots, n\}$
- Compute PCE $k(\boldsymbol{\xi}) = \sum k_\beta \psi_\beta(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi}) = \sum \varphi_\gamma \psi_\gamma(\boldsymbol{\xi})$ using $\{k_i, \phi_i, i = 1, \ldots, n\}$



Stochastic time transform

Time-transform PCE

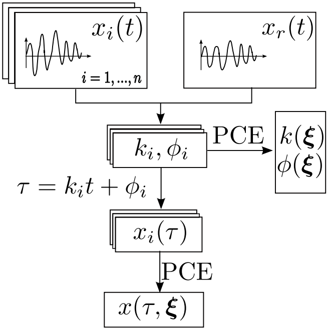
- Choose a reference trajectory $x_r(t)$
- Define a linear stochastic time transform $\tau = k\,t + \phi$
- For $i = 1, \ldots, n$,
  - Determine $\{k_i, \phi_i\}$ as solution of an optimization problem.
  - Represent $x_i(t)$ on the transformed time line τ, yielding $x_i(\tau)$.
- Compute PCE $x(\tau, \boldsymbol{\xi}) = \sum x_{\alpha}(\tau)\psi_{\alpha}(\boldsymbol{\xi})$ using $\{x_i(\tau, \boldsymbol{\xi}), i = 1, \ldots, n\}$
- Compute PCE $k(\boldsymbol{\xi}) = \sum k_{\beta}\psi_{\beta}(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi}) = \sum \varphi_{\gamma}\psi_{\gamma}(\boldsymbol{\xi})$ using $\{k_i, \phi_i, i = 1, \ldots, n\}$



Stochastic time transform

Time-transform PCE

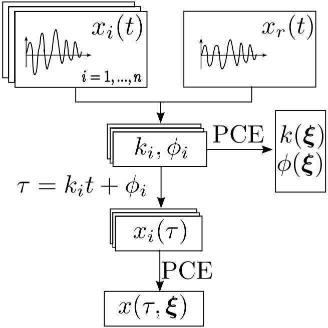
- Choose a reference trajectory $x_r(t)$
- Define a linear stochastic time transform $\tau = k\,t + \phi$
- For $i = 1, \ldots, n$,
  - Determine $\{k_i, \phi_i\}$ as solution of an optimization problem.
  - Represent $x_i(t)$ on the transformed time line τ, yielding $x_i(\tau)$.
- Compute PCE $x(\tau, \boldsymbol{\xi}) = \sum x_{\boldsymbol{\alpha}}(\tau)\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ using $\{x_i(\tau, \boldsymbol{\xi}), i = 1, \ldots, n\}$
- Compute PCE $k(\boldsymbol{\xi}) = \sum k_\beta \psi_\beta(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi}) = \sum \varphi_\gamma \psi_\gamma(\boldsymbol{\xi})$ using $\{k_i, \phi_i, i = 1, \ldots, n\}$



Stochastic time transform

Time-transform PCE

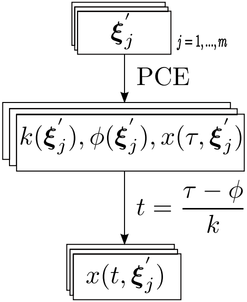
- Choose a reference trajectory $x_r(t)$
- Define a linear stochastic time transform $\tau = k\,t + \phi$
- For $i = 1, \ldots, n$,
  - Determine $\{k_i, \phi_i\}$ as solution of an optimization problem.
  - Represent $x_i(t)$ on the transformed time line τ, yielding $x_i(\tau)$.
- Compute PCE $x(\tau, \boldsymbol{\xi}) = \sum x_{\boldsymbol{\alpha}}(\tau)\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ using $\{x_i(\tau, \boldsymbol{\xi}), i = 1, \ldots, n\}$
- Compute PCE $k(\boldsymbol{\xi}) = \sum k_{\boldsymbol{\beta}}\psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi}) = \sum \varphi_{\boldsymbol{\gamma}}\psi_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ using $\{k_i, \phi_i, i = 1, \ldots, n\}$



Stochastic time transform

Prediction by time-transform PCE

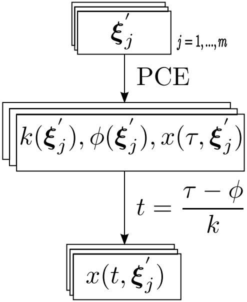
- Predict $x(\tau, \boldsymbol{\xi}')$, $k(\boldsymbol{\xi}')$ and $\phi(\boldsymbol{\xi}')$ using the computed PCE.
- Map $x(\tau, \boldsymbol{\xi}')$ into $x(t, \boldsymbol{\xi}')$ using $t = \dfrac{\tau - \phi(\boldsymbol{\xi}')}{k(\boldsymbol{\xi}')}$



Stochastic time transform

Prediction by time-transform PCE

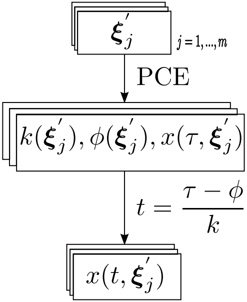
- Predict $x(\tau, \boldsymbol{\xi}')$, $k(\boldsymbol{\xi}')$ and $\phi(\boldsymbol{\xi}')$ using the computed PCE.
- Map $x(\tau, \boldsymbol{\xi}')$ into $x(t, \boldsymbol{\xi}')$ using $t = \dfrac{\tau - \phi(\boldsymbol{\xi}')}{k(\boldsymbol{\xi}')}$



Stochastic time transform

Prediction by time-transform PCE

- Predict $x(\tau, \boldsymbol{\xi}')$, $k(\boldsymbol{\xi}')$ and $\phi(\boldsymbol{\xi}')$ using the computed PCE.
- Map $x(\tau, \boldsymbol{\xi}')$ into $x(t, \boldsymbol{\xi}')$ using $t = \dfrac{\tau - \phi(\boldsymbol{\xi}')}{k(\boldsymbol{\xi}')}$



Optimization-based time transform

Maximize the similarity measure between the considered trajectory and the reference one

$$(k, \phi) = \arg \max_{\substack{k \in \mathcal{R}^+ \\ |\phi| \leqslant T_r/4}} g(k, \phi)$$

Optimization-based time transform

Maximize the similarity measure between the considered trajectory and the reference one

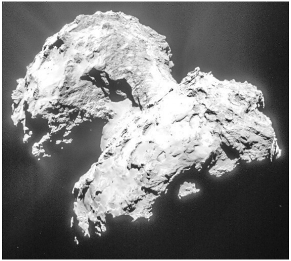
$$(k, \phi) = \arg \max_{\substack{k \in \mathcal{R}^+ \\ |\phi| \leqslant T_r/4}} g(k, \phi)$$

where

$$g(k, \phi) = \frac{\left| \int_0^T x(k\,t + \phi) x_r(t) \mathrm{d}t \right|}{\| x(k\,t + \phi) \| \| x_r(t) \|}$$

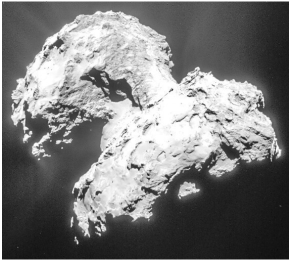
Outline

Rigid body dynamics



http://www.esa.int/spaceinimages/Missions/Rosetta

Rigid body dynamics

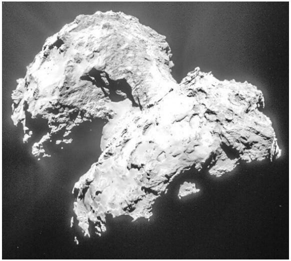


http://www.esa.int/spaceinimages/Missions/Rosetta

Rotation of a rigid body described by Euler's equations

$$\begin{cases} M_x = I_{xx}\,\dot{\omega}_x - (I_{yy} - I_{zz})\,\omega_y\,\omega_z \\ M_y = I_{yy}\,\dot{\omega}_y - (I_{zz} - I_{xx})\,\omega_z\,\omega_x \\ M_z = I_{zz}\,\dot{\omega}_z - (I_{xx} - I_{yy})\,\omega_x\,\omega_y \end{cases}$$

Rigid body dynamics



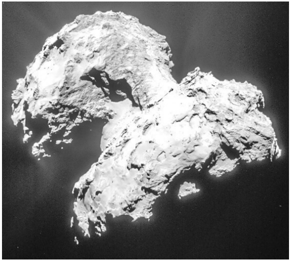
http://www.esa.int/spaceinimages/Missions/Rosetta

Rotation of a rigid body described by Euler's equations

$$\begin{cases} M_x = I_{xx}\,\dot{\omega}_x - (I_{yy} - I_{zz})\,\omega_y\,\omega_z \\ M_y = I_{yy}\,\dot{\omega}_y - (I_{zz} - I_{xx})\,\omega_z\,\omega_x \\ M_z = I_{zz}\,\dot{\omega}_z - (I_{xx} - I_{yy})\,\omega_x\,\omega_y \end{cases}$$

- $M_x = M_y = M_z = 0$
- $x(0) = 0,\ y(0) = 1,\ z(0) = 1$
- $I_{xx} = \dfrac{1-c}{2} I_{yy},\ I_{zz} = \dfrac{1+c}{2} I_{yy}$

Rigid body dynamics



http://www.esa.int/spaceinimages/Missions/Rosetta

Rotation of a rigid body described by Euler's equations

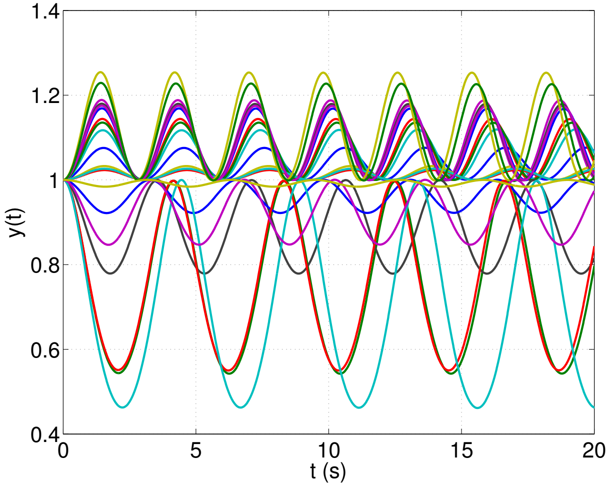
$$\begin{cases} M_x = I_{xx}\,\dot{\omega}_x - (I_{yy} - I_{zz})\,\omega_y\,\omega_z \\ M_y = I_{yy}\,\dot{\omega}_y - (I_{zz} - I_{xx})\,\omega_z\,\omega_x \\ M_z = I_{zz}\,\dot{\omega}_z - (I_{xx} - I_{yy})\,\omega_x\,\omega_y \end{cases}$$

- $M_x = M_y = M_z = 0$
- $x(0) = 0,\ y(0) = 1,\ z(0) = 1$
- $I_{xx} = \dfrac{1-c}{2} I_{yy},\ I_{zz} = \dfrac{1+c}{2} I_{yy}$

$$\begin{cases} \dot{x} = yz \\ \dot{y} = c \times xz \\ \dot{z} = -xy \end{cases}$$

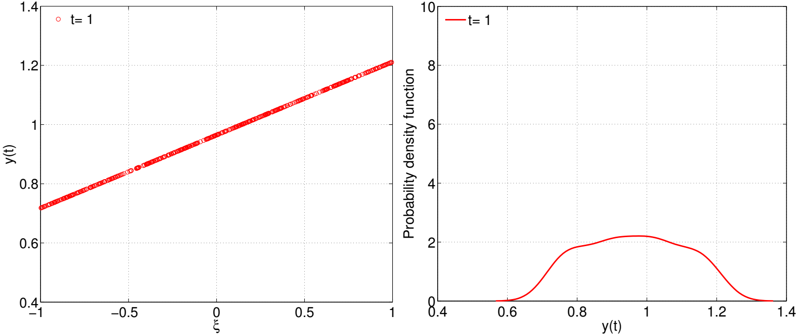
$$c = 0.1(7 \times \xi - 1);$$
$$\xi \sim \mathcal{U}(-1, 1)$$

PCE for rigid body dynamics



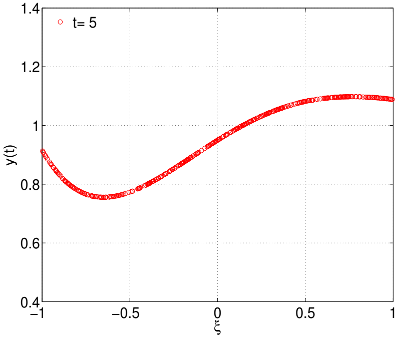
Distinct response trajectories in the original time t

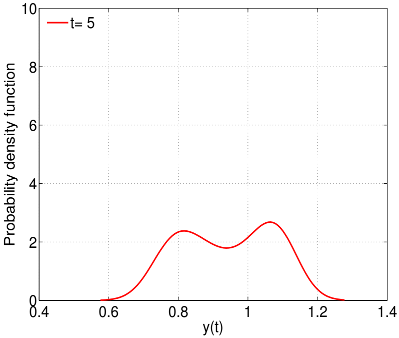
PCE for rigid body dynamics



Input-output relationship in t and output's PDF

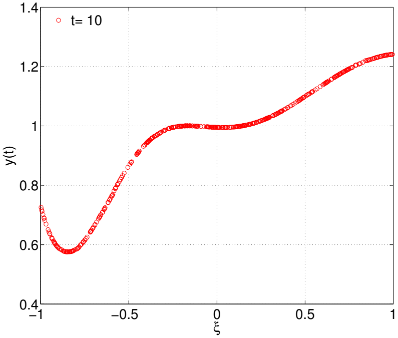
PCE for rigid body dynamics

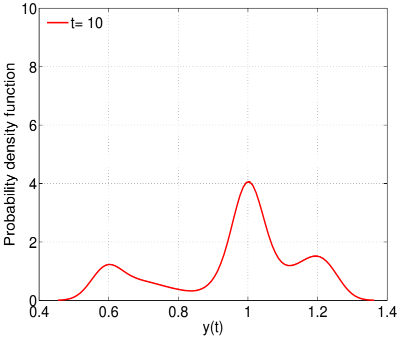




Input-output relationship in t and output's PDF

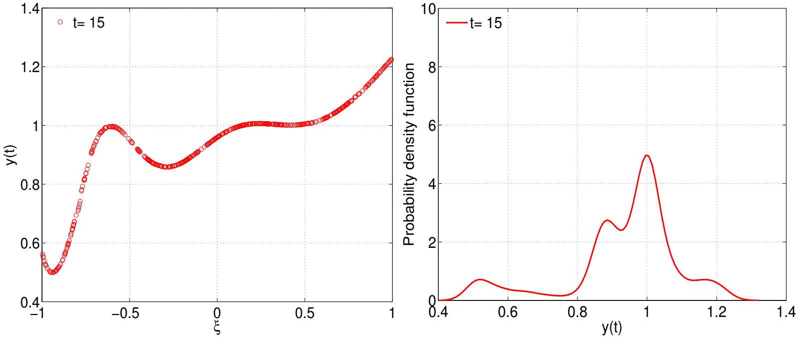
PCE for rigid body dynamics





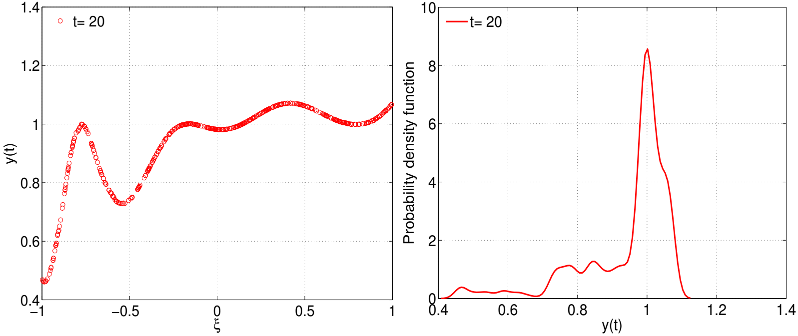
Input-output relationship in t and output's PDF

PCE for rigid body dynamics



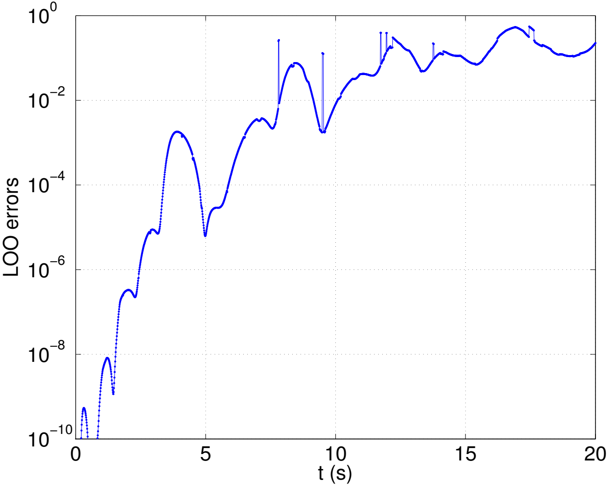
Input-output relationship in t and output's PDF

PCE for rigid body dynamics



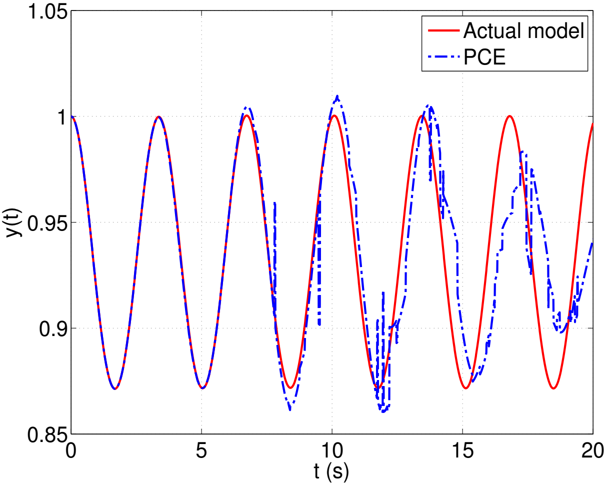
Input-output relationship in t and output's PDF

PCE prediction *vs.* actual response trajectory



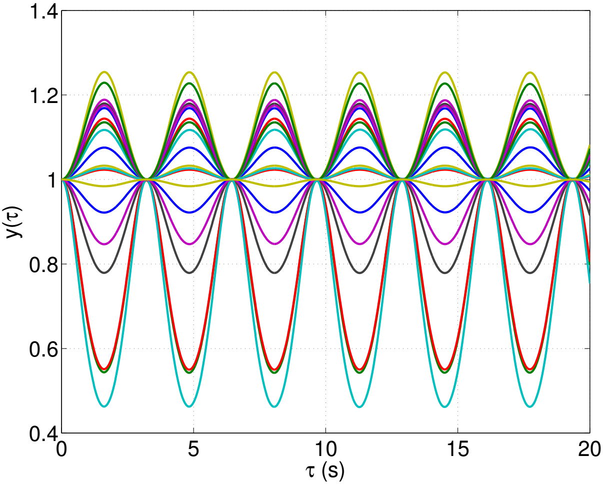
LOO errors *vs.* time t

PCE prediction *vs.* actual response trajectory



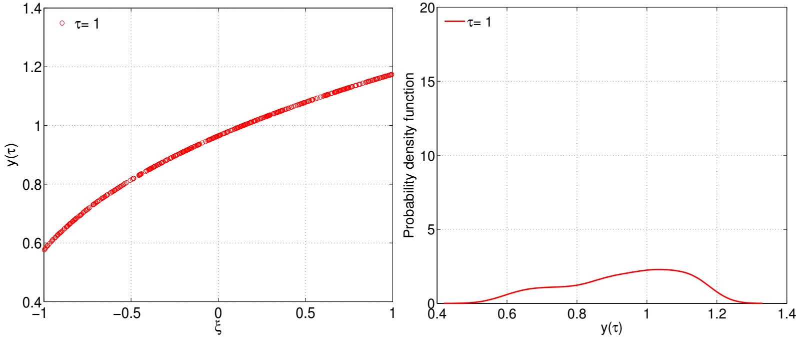
Frozen-in-time PCE *vs.* numerical model

PCE for rigid body dynamics



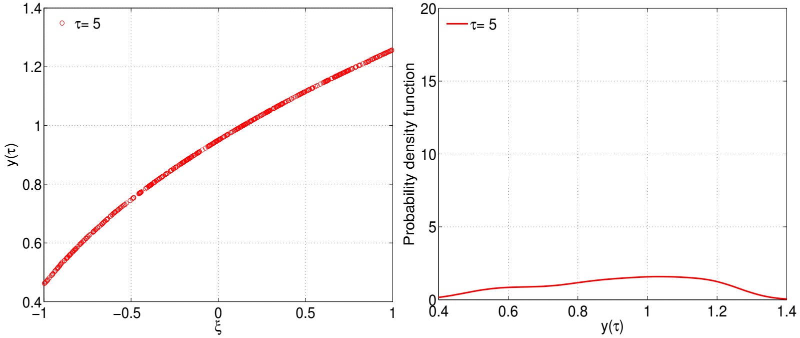
Distinct response trajectories in the time τ

PCE for rigid body dynamics



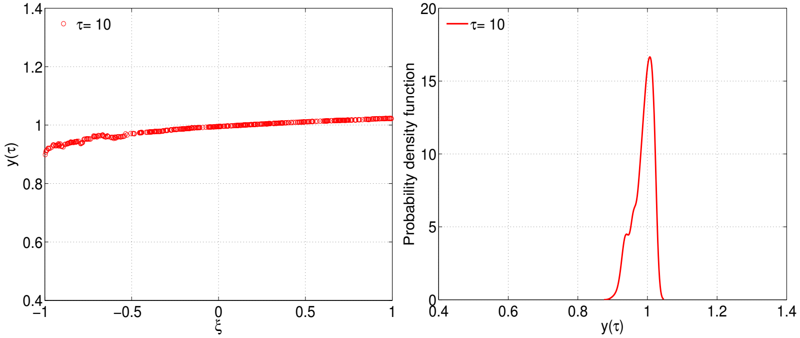
Input-output relationship in τ and output's PDF

PCE for rigid body dynamics



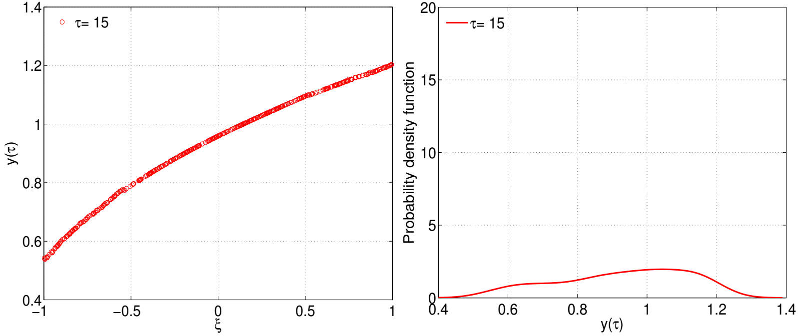
Input-output relationship in τ and output's PDF

PCE for rigid body dynamics



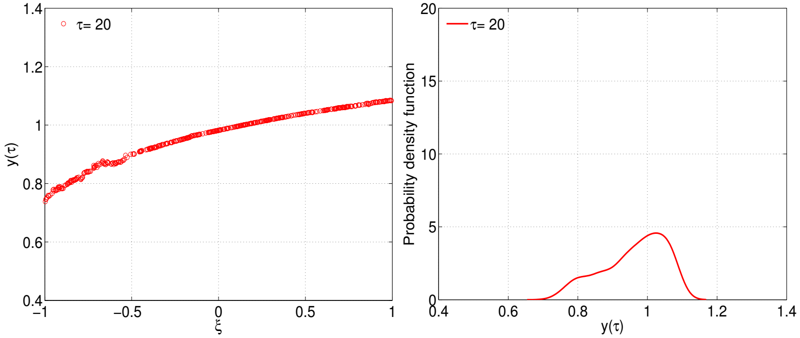
Input-output relationship in τ and output's PDF

PCE for rigid body dynamics



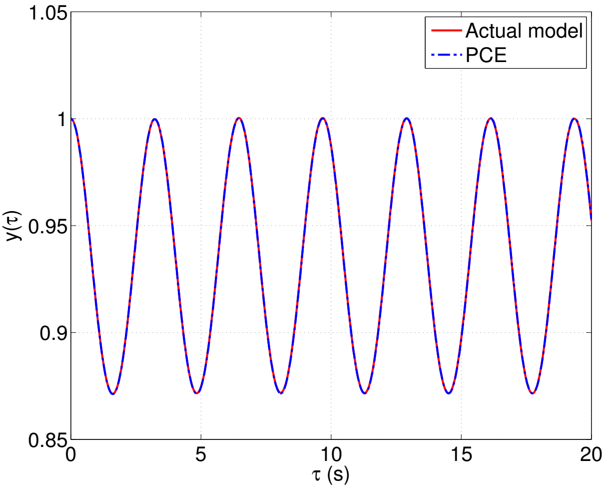
Input-output relationship in τ and output's PDF

PCE for rigid body dynamics



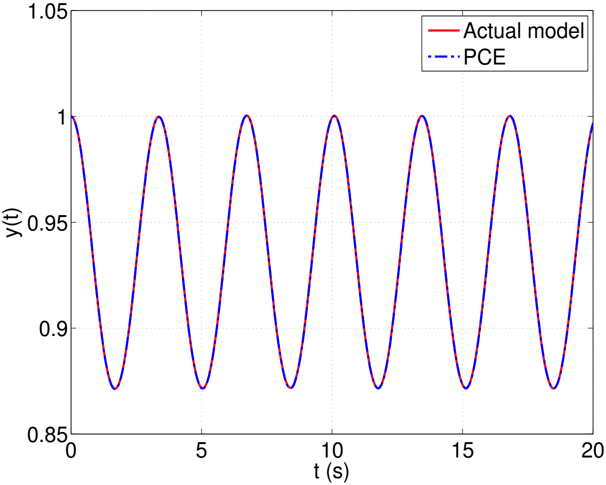
Input-output relationship in τ and output's PDF

PCE prediction *vs.* actual response trajectory



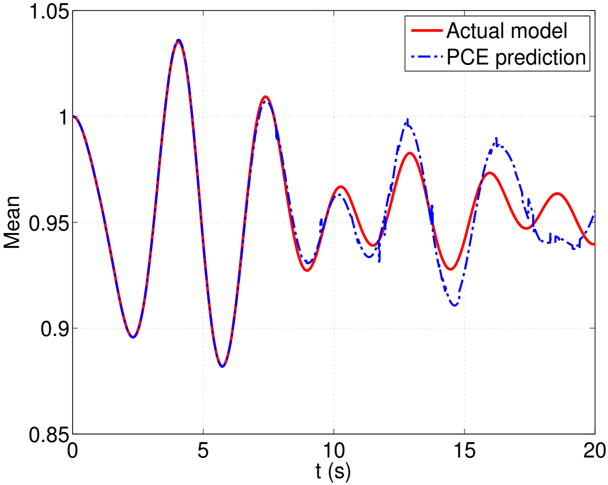
in the transformed time τ

PCE prediction *vs.* actual response trajectory



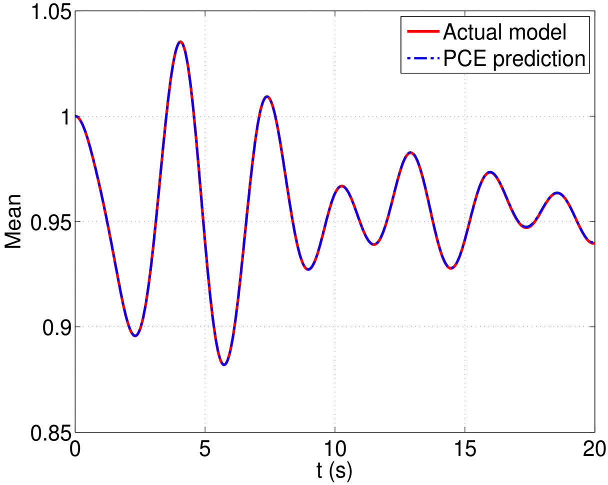
in the original time t

PCE *vs.* MCS



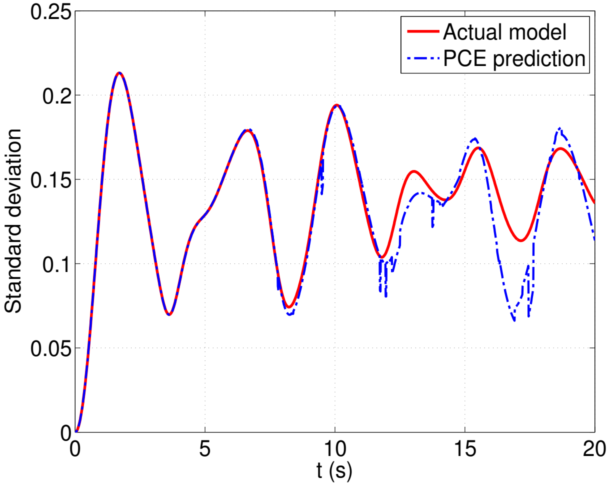
Frozen-in-time PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



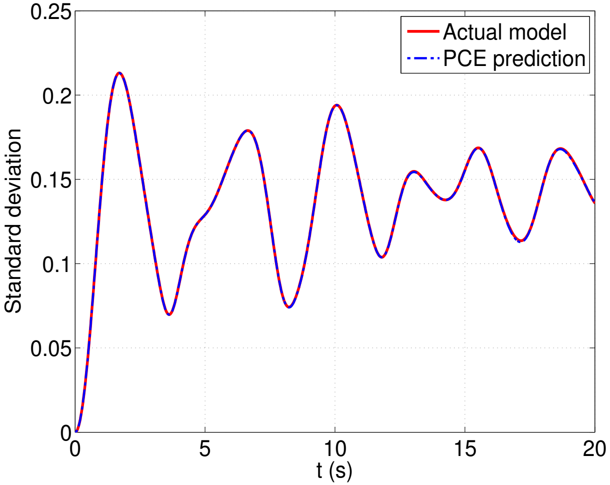
Time-transform PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



Frozen-in-time PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



Time-transform PCE *vs.* MCS (10000 runs)

Duffing oscillator

3d-pictures.picphotos.net

Duffing oscillator

3d-pictures.picphotos.net

Non-linear SDOF Duffing oscillator:

$$\ddot{x}(t) + 2\,\omega\,\zeta\,\dot{x}(t) + \omega^2\,(x(t) + \epsilon\,x^3(t)) = 0$$

where

- $\zeta = 0.05(1 + 0.05\xi_1), \quad \xi_1 \sim \mathcal{U}(-1, 1)$
- $\omega = 2\pi(1 + 0.2\xi_2), \quad \xi_2 \sim \mathcal{U}(-1, 1)$
- $\epsilon = -0.5(1 + 0.5\xi_3), \quad \xi_3 \sim \mathcal{U}(-1, 1)$
- $x(t = 0) = 1$ and $\dot{x}(t = 0) = 0$.

Duffing oscillator

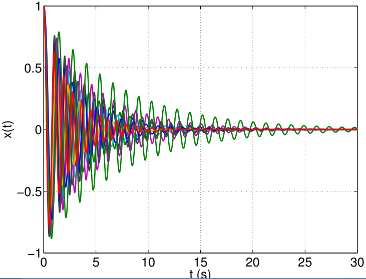
3d-pictures.picphotos.net

Non-linear SDOF Duffing oscillator:

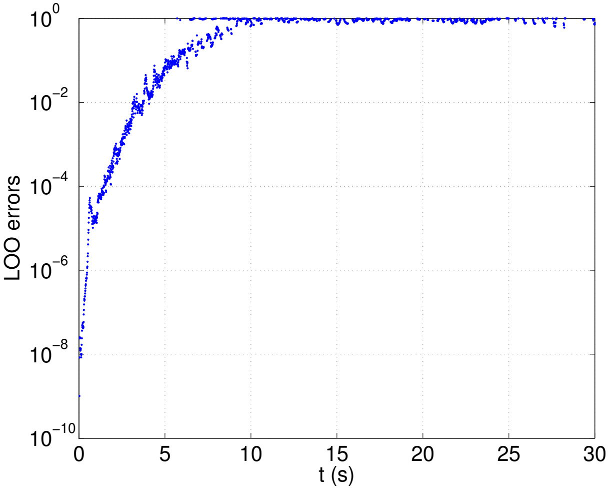
$$\ddot{x}(t) + 2\,\omega\,\zeta\,\dot{x}(t) + \omega^2\,(x(t) + \epsilon\,x^3(t)) = 0$$

where

- $\zeta = 0.05(1 + 0.05\xi_1), \quad \xi_1 \sim \mathcal{U}(-1, 1)$
- $\omega = 2\pi(1 + 0.2\xi_2), \quad \xi_2 \sim \mathcal{U}(-1, 1)$
- $\epsilon = -0.5(1 + 0.5\xi_3), \quad \xi_3 \sim \mathcal{U}(-1, 1)$
- $x(t = 0) = 1$ and $\dot{x}(t = 0) = 0$.

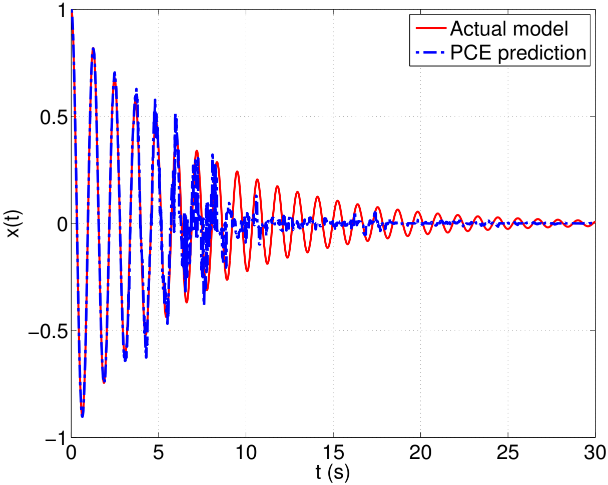


PCE for Duffing oscillator



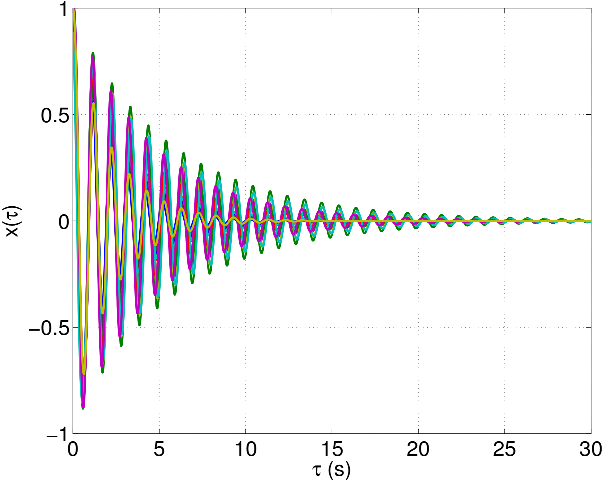
LOO errors *vs.* time

PCE for Duffing oscillator



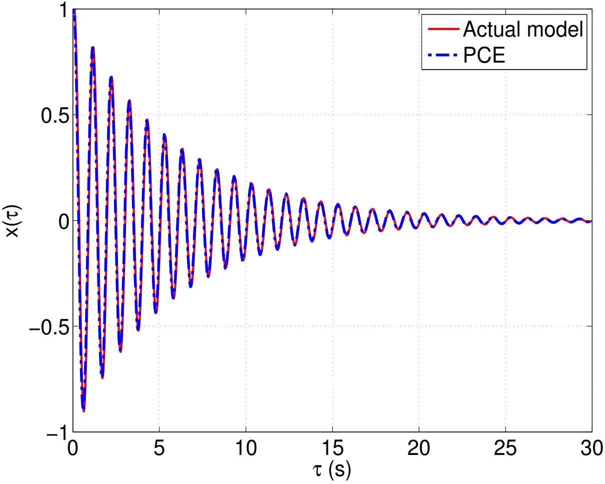
Frozen-in-time PCE *vs.* numerical model in time t

PCE for Duffing oscillator



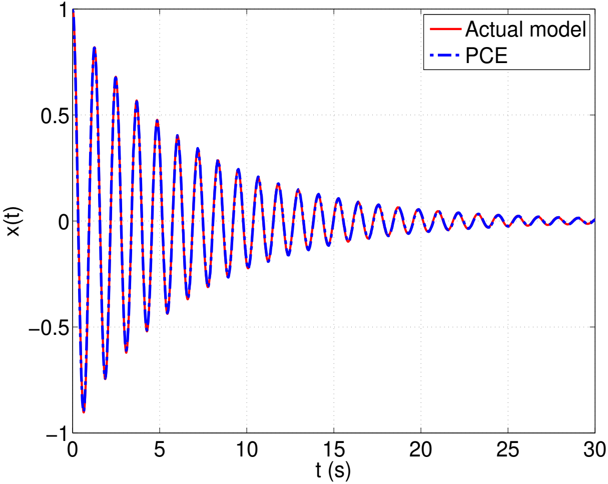
Distinct responses in the transformed time τ

PCE for Duffing oscillator



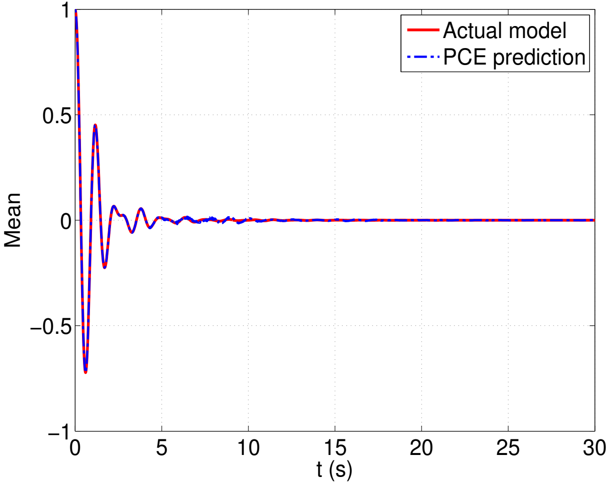
in time τ

PCE for Duffing oscillator



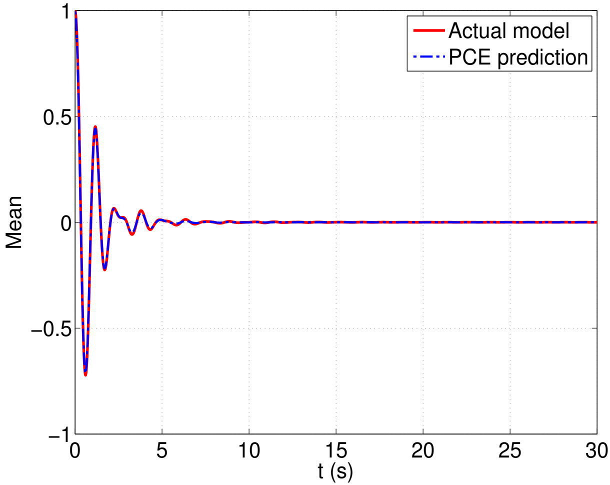
in time t

PCE *vs.* MCS



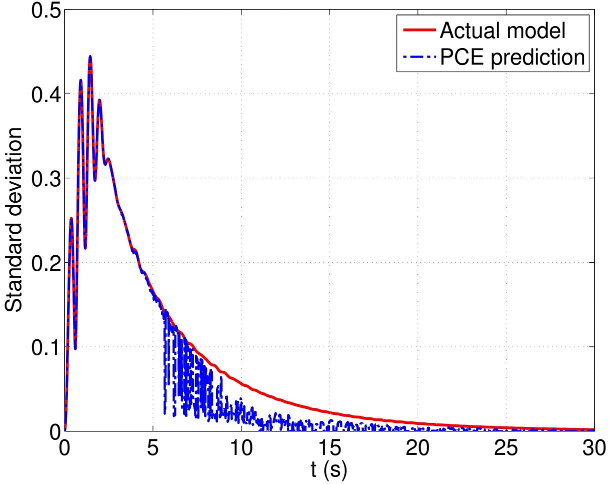
Frozen-in-time PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



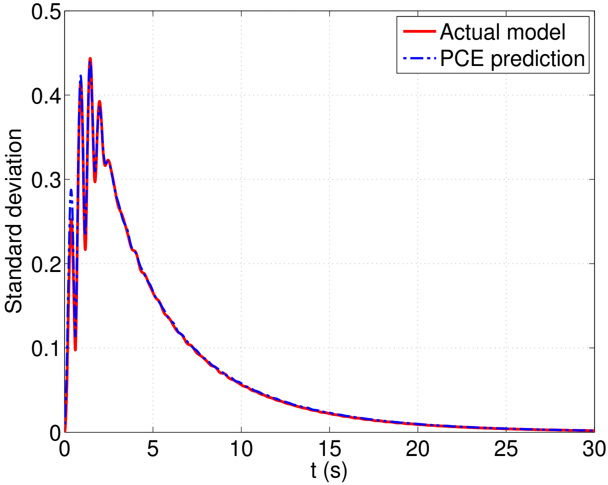
Time-transform PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



Frozen-in-time PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



Time-transform PCE *vs.* MCS (10000 runs)

Oregonator model

Le Maître, Mathelin, et al., 2010

The Oregonator model describes the dynamics of a well-stirred, homogeneous chemical system governed by a three species coupled mechanism:

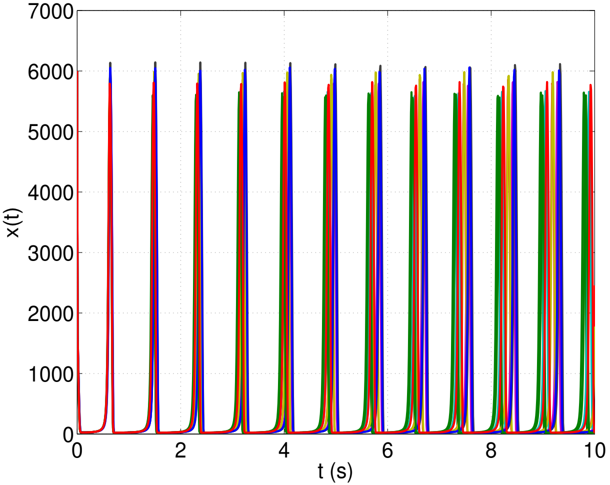
$$\begin{cases} \dot{x} = k_1\, y(t) - k_2\, x(t)\, y(t) + k_3\, x(t) - k_4\, x(t)^2 \\ \dot{y} = -k_1\, y(t) - k_2\, x(t)\, y(t) + k_5\, z(t) \\ \dot{z} = k_3\, x(t) - k_5\, z(t) \end{cases}$$

dreamstime.com

in which

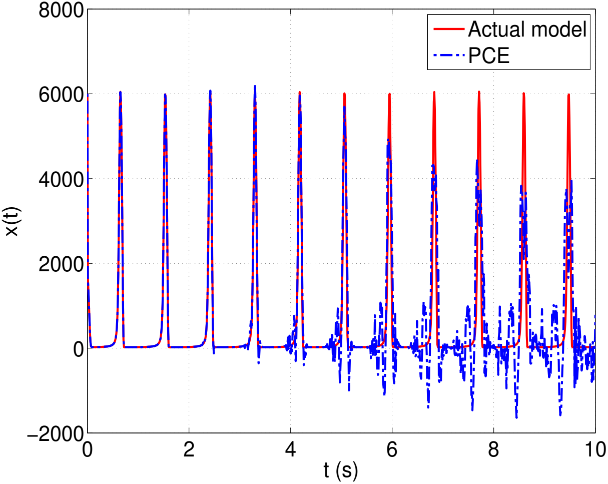
- (x, y, z) indicates the three species concentration
- the initial condition $(x_0, y_0, z_0) = (6000, 6000, 6000)$ corresponds to a deterministic mixture.
- $k_i,\ i = 1, \ldots, 5$ are the reaction parameters.
  - $k_1 = 2,\ k_2 = 0.1,\ k_3 = 104$
  - $k_4 = 0.008(1 + 0.05\xi_1), \qquad \xi_1 \sim \mathcal{U}(-1, 1)$
  - $k_5 = 26(1 + 0.1\xi_2), \qquad \xi_2 \sim \mathcal{U}(-1, 1)$

PCE for Oregonator model



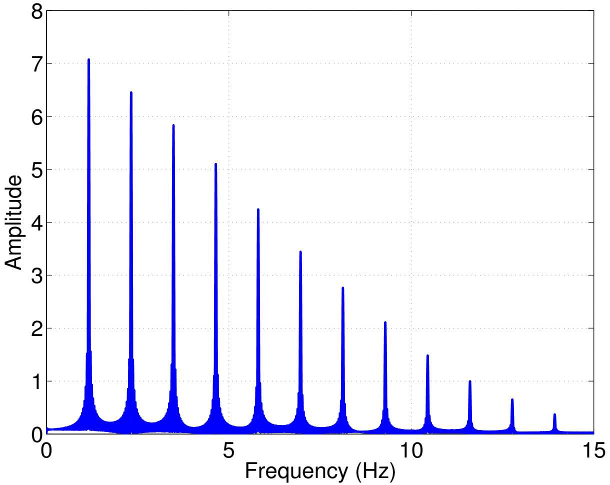
Responses in t

PCE for Oregonator model



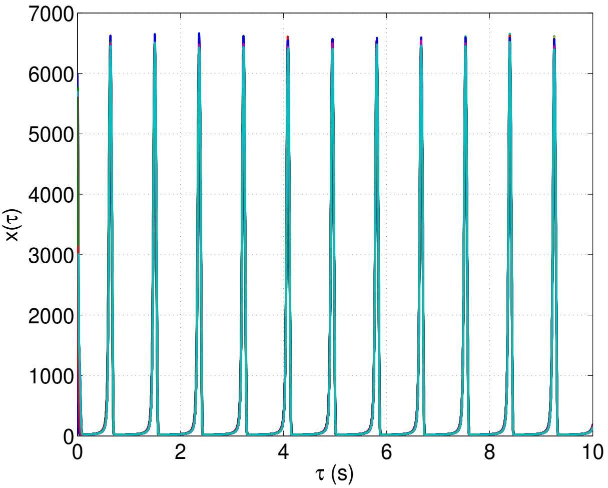
PCE prediction *vs.* actual response in t

PCE for Oregonator model



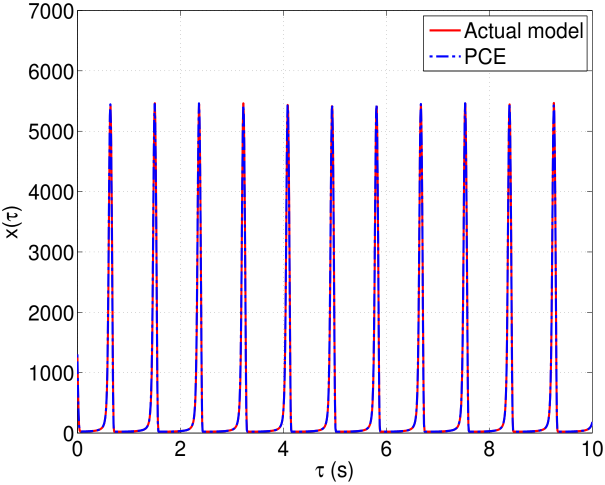
Amplitude spectrum of an output trajectory

PCE for Oregonator model



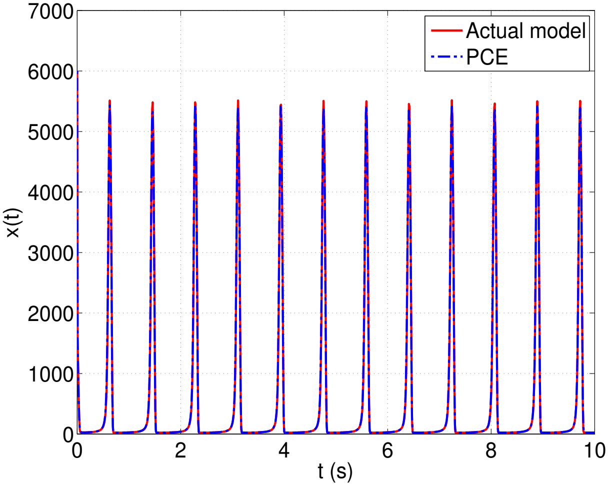
Responses in τ

PCE prediction *vs.* actual response trajectory



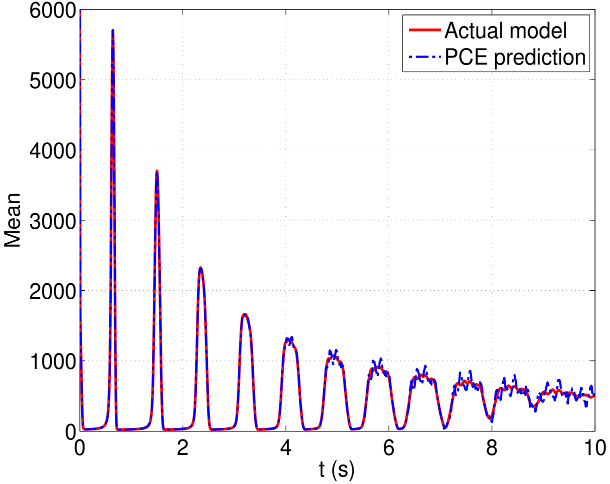
in the transformed time τ

PCE prediction *vs.* actual response trajectory



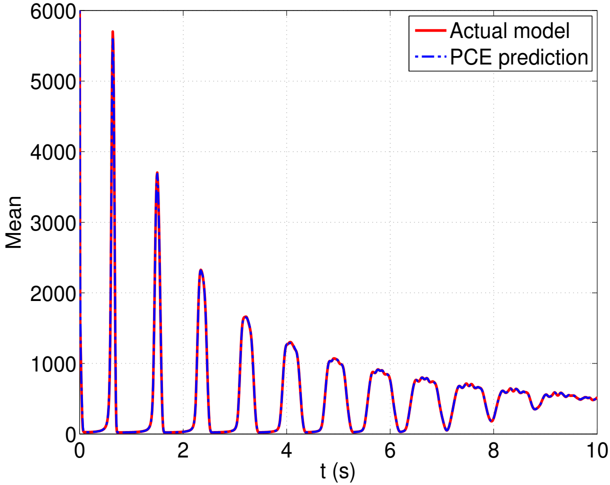
in the original time t

PCE *vs.* MCS



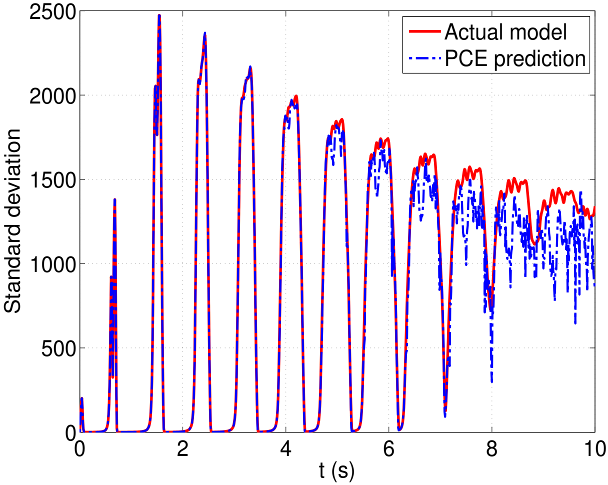
Frozen-in-time PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



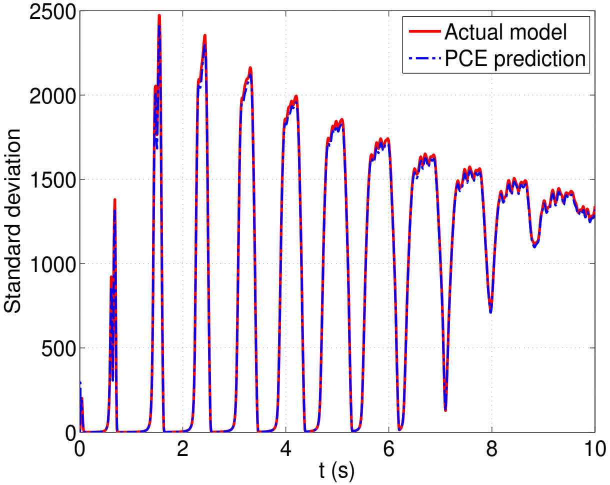
Time-transform PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



Frozen-in-time PCE *vs.* MCS (10000 runs)

PCE *vs.* MCS



Time-transform PCE *vs.* MCS (10000 runs)

Outline

Conclusions and perspective

Conclusions

- Polynomial chaos expansions are facing challenging issues when modelling time-dependent systems.
- A non-intrusive stochastic time transform approach is proposed to solve the problems of non-linear oscillators.
- The approach is proved effective in some examples.

Perspective

- Further investigation is required to extend the approach to more complex problems involving output signals which are rich in the frequency content.

Conclusions and perspective

Conclusions

- Polynomial chaos expansions are facing challenging issues when modelling time-dependent systems.
- A non-intrusive stochastic time transform approach is proposed to solve the problems of non-linear oscillators.
- The approach is proved effective in some examples.

Perspective

- Further investigation is required to extend the approach to more complex problems involving output signals which are rich in the frequency content.

Conclusions and perspective

Conclusions

- Polynomial chaos expansions are facing challenging issues when modelling time-dependent systems.
- A non-intrusive stochastic time transform approach is proposed to solve the problems of non-linear oscillators.
- The approach is proved effective in some examples.

Perspective

- Further investigation is required to extend the approach to more complex problems involving output signals which are rich in the frequency content.

Conclusions and perspective

Conclusions

- Polynomial chaos expansions are facing challenging issues when modelling time-dependent systems.
- A non-intrusive stochastic time transform approach is proposed to solve the problems of non-linear oscillators.
- The approach is proved effective in some examples.

Perspective

- Further investigation is required to extend the approach to more complex problems involving output signals which are rich in the frequency content.

Questions ?

Thank you very much for your attention !

Chair of Risk, Safety & Uncertainty Quantification

`http://www.rsuq.ethz.ch`

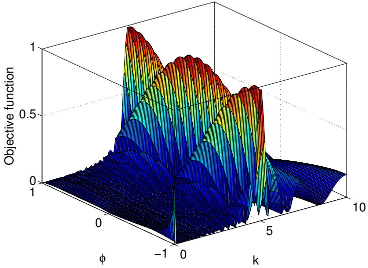
UQLAB ...

... The Uncertainty Quantification Laboratory

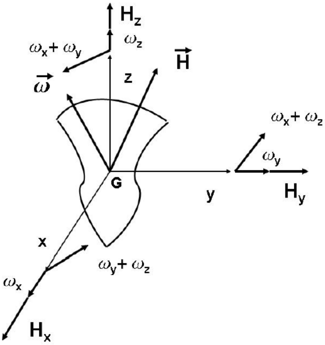
Backup slides

Optimization-based time transform

The constraint on the support of the parameters is adopted to ensure the uniqueness of the solution.

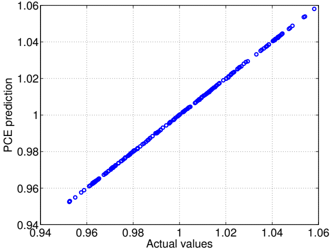


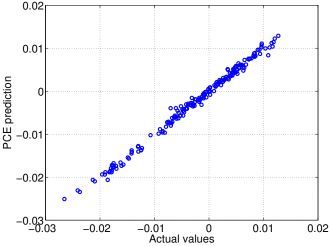
Rigid body dynamics



Rigid body dynamics

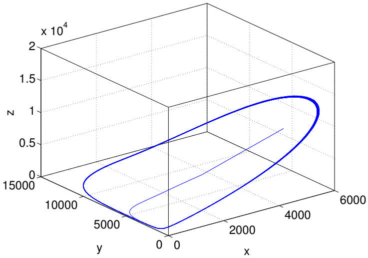
PCE for Oregonator model





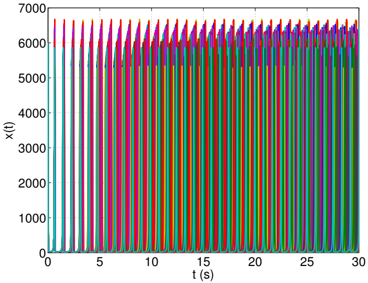
Prediction of k (left) and ϕ (right) by PCE *vs.* actual values

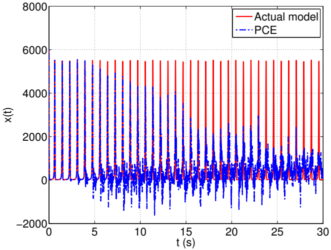
Oregonator model

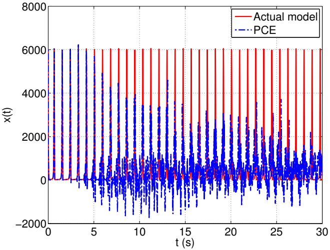


Limit cycle oscillation

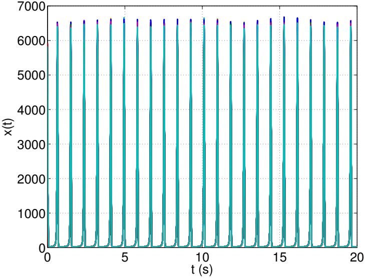
PCE for Oregonator model

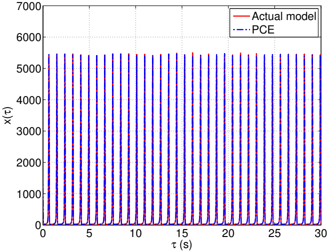


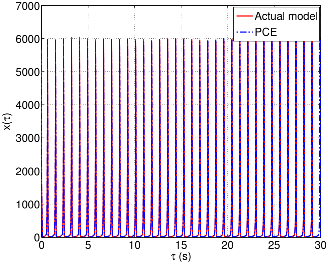




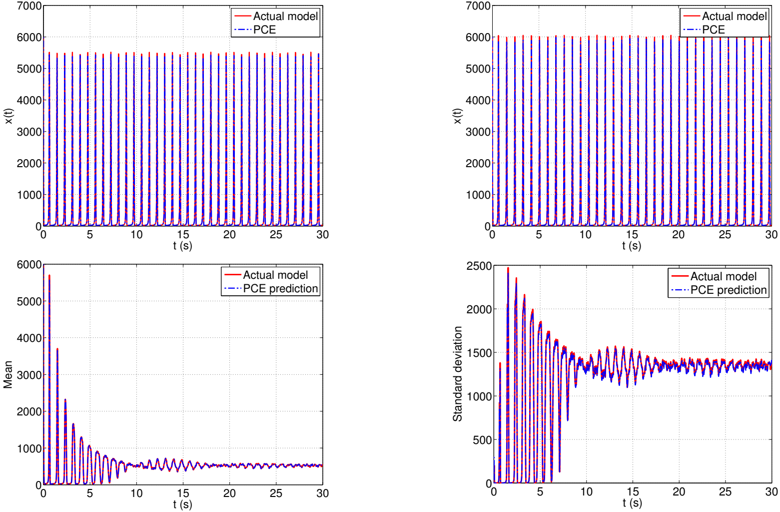
PCE for Oregonator model



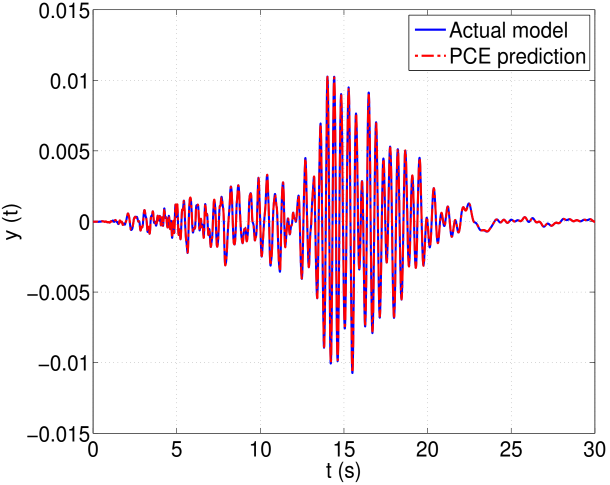




PCE for Oregonator model



PCE for seismic analysis



First story displacement of a non-linear steel frame subject to random seismic excitation