Surrogate models for uncertain dynamical systems: polynomial chaos expansions for time-dependent responses

Chu V. Mai, Bruno Sudret

Chair of Risk, Safety and Uncertainty Quantification

ETH Zurich

4ème Forum DAM "Méthodes de Quantification des Incertitudes"
4 Octobre 2016

Chair of Risk, Safety and Uncertainty quantification

The Chair carries out research projects in the field of uncertainty quantification for engineering problems with applications in structural reliability, sensitivity analysis, model calibration and reliability-based design optimization

Research topics

- Uncertainty modelling for engineering systems
- Structural reliability analysis
- Surrogate models (polynomial chaos expansions, Kriging, support vector machines)
- Bayesian model calibration and stochastic inverse problems
- Global sensitivity analysis
- Reliability-based design optimization

http://www.rsuq.ethz.ch

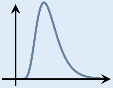
Global framework for uncertainty quantification

Step B
Quantification of sources of uncertainty

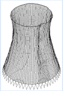
Step A
Model(s) of the system
Assessment criteria

Step C
Uncertainty propagation

Random variables



Computational model



Moments
Probability of failure
Response PDF

Step C'
Sensitivity analysis

B. Sudret, *Uncertainty propagation and sensitivity analysis in mechanical models – contributions to structural reliability and stochastic spectral methods (2007)*

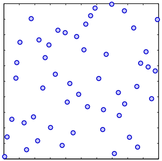
Surrogate models for uncertainty quantification

A surrogate model $\tilde{\mathcal{M}}$ is an approximation of the original computational model $\mathcal{M}$ with the following features:

- It is built from a limited set of runs of the original model $\mathcal{M}$ called the experimental design $\mathcal{X} = \left\{ \boldsymbol{x}^{(i)},\, i = 1, \ldots, n \right\}$
- It assumes some regularity of the model $\mathcal{M}$ and some general functional shape

Name	Shape	Parameters
Polynomial chaos expansions	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$	$\boldsymbol{a}_{\boldsymbol{\alpha}}$
Low-rank tensor approximations	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} v_l^{(i)}(x_i) \right)$	$b_l,\, z_{k,l}^{(i)}$
Kriging (a.k.a Gaussian processes)	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) + Z(\boldsymbol{x}, \omega)$	$\boldsymbol{\beta}\,,\, \sigma_Z^2\,,\, \boldsymbol{\theta}$
Support vector machines	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{n} a_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\boldsymbol{a}\,,\, b$

Ingredients for building a surrogate model



- Select an experimental design $\mathcal{X}$ that covers at best the domain of input parameters: Latin hypercube sampling (LHS), low-discrepancy sequences
- Run the computational model $\mathcal{M}$ onto $\mathcal{X}$ exactly as in Monte Carlo simulation
- Smartly post-process the data $\{\mathcal{X}, \mathcal{M}(\mathcal{X})\}$ through a learning algorithm

Name	**Learning method**
Polynomial chaos expansions	sparse grid integration, least-squares, compressive sensing
Low-rank tensor approximations	alternate least squares
Kriging	maximum likelihood, Bayesian inference
Support vector machines	quadratic programming

Advantages of surrogate models

Usage

$$\mathcal{M}(\boldsymbol{x}) \quad \approx \quad \tilde{\mathcal{M}}(\boldsymbol{x})$$

hours per run — seconds for 10^6 runs

Advantages

- Non-intrusive methods: based on runs of the computational model, exactly as in Monte Carlo simulation
- Construction suited to high performance computing: "embarrassingly parallel"

Challenges

- Need for rigorous validation
- Communication: advanced mathematical background

Efficiency: 2-3 orders of magnitude less runs compared to Monte Carlo

Outline

1. Introduction

2. Polynomial chaos expansions
   - Polynomial chaos basis
   - Computing the PCE coefficients

3. Time-warping PCE
   - Introduction
   - Stochastic time warping
   - Oregonator model
   - Bouc-Wen model

4. PC-NARX expansions
   - NARX model
   - Calibration of a PC-NARX model
   - Application to Bouc Wen model

Polynomial chaos expansions in a nutshell

Ghanem & Spanos (1991); Sudret & Der Kiureghian (2000); Xiu & Karniadakis (2002); Soize & Ghanem (2004)

- Consider the input random vector $\boldsymbol{X}$ (dim $\boldsymbol{X} = M$) with given probability density function (PDF) $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{M} f_{X_i}(x_i)$
- Assuming that the random output $Y = \mathcal{M}(\boldsymbol{X})$ has finite variance, it can be cast as the following polynomial chaos expansion:

$$Y = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} y_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

  where :

  - $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$: basis functions
  - $y_{\boldsymbol{\alpha}}$: coefficients to be computed (coordinates)

- The PCE basis $\left\{ \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \, \boldsymbol{\alpha} \in \mathbb{N}^M \right\}$ is made of multivariate orthonormal polynomials

Multivariate polynomial basis

Univariate polynomials

- For each input variable X_i, univariate orthogonal polynomials $\{P_k^{(i)},\ k \in \mathbb{N}\}$ are built:

$$\left\langle P_j^{(i)}, P_k^{(i)} \right\rangle = \int P_j^{(i)}(u)\ P_k^{(i)}(u)\ f_{X_i}(u)\, du =\ \gamma_j^{(i)}\ \delta_{jk}$$

e.g. , Legendre polynomials if $X_i \sim \mathcal{U}(-1, 1)$, Hermite polynomials if $X_i \sim \mathcal{N}(0, 1)$

- Normalization: $\Psi_j^{(i)} = P_j^{(i)} / \sqrt{\gamma_j^{(i)}} \quad i = 1, \ldots, M, \quad j \in \mathbb{N}$

Tensor product construction

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \prod_{i=1}^{M} \Psi_{\alpha_i}^{(i)}(x_i) \qquad \mathbb{E}\left[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})\Psi_{\boldsymbol{\beta}}(\boldsymbol{X})\right] = \delta_{\boldsymbol{\alpha\beta}}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ are multi-indices (partial degree in each dimension)

Outline

Computing the coefficients by least-square minimization

Isukapalli (1999); Berveiller, Sudret & Lemaire (2006)

Principle

The exact (infinite) series expansion is considered as the sum of a truncated series and a residual:

$$Y = \mathcal{M}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) + \varepsilon_P \equiv \mathbf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{X}) + \varepsilon_P(\boldsymbol{X})$$

where : $\mathbf{Y} = \{y_{\boldsymbol{\alpha}}, \, \boldsymbol{\alpha} \in \mathcal{A}\} \equiv \{y_0, \, \dots \, , y_{P-1}\}$ (P unknown coef.)

$$\boldsymbol{\Psi}(\boldsymbol{x}) = \{\Psi_0(\boldsymbol{x}), \, \dots \, , \Psi_{P-1}(\boldsymbol{x})\}$$

Least-square minimization

The unknown coefficients are estimated by minimizing the mean square residual error:

$$\hat{\mathbf{Y}} = \arg\min \mathbb{E}\left[\left(\mathbf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{X}) - \mathcal{M}(\boldsymbol{X})\right)^2\right]$$

Discrete (ordinary) least-square minimization

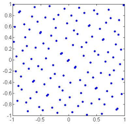
An estimate of the mean square error (sample average) is minimized:

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}\in\mathbb{R}^P} \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{Y}^\mathsf{T}\boldsymbol{\Psi}(\boldsymbol{x}^{(i)}) - \mathcal{M}(\boldsymbol{x}^{(i)})\right)^2$$

Procedure

- Select a truncation scheme, *e.g.* $\mathcal{A}^{M,p} = \left\{\boldsymbol{\alpha}\in\mathbb{N}^M \ : \ |\boldsymbol{\alpha}|_1 \leq p\right\}$
- Select an experimental design and evaluate the model response

$$\mathbf{M} = \left\{\mathcal{M}(\boldsymbol{x}^{(1)}), \dots, \mathcal{M}(\boldsymbol{x}^{(n)})\right\}^\mathsf{T}$$



- Compute the experimental matrix

$$\mathbf{A}_{ij} = \Psi_j\left(\boldsymbol{x}^{(i)}\right) \quad i = 1, \dots, n \; ; \; j = 0, \dots, P-1$$

- Solve the resulting linear system

$$\hat{\mathbf{Y}} = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{M}$$

Simple is beautiful !

Error estimators

- In least-squares analysis, the generalization error is defined as:

$$E_{gen} = \mathbb{E}\left[\left(\mathcal{M}(\boldsymbol{X}) - \mathcal{M}^{\mathsf{PC}}(\boldsymbol{X})\right)^2\right] \qquad \mathcal{M}^{\mathsf{PC}}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

- The empirical error based on the experimental design $\mathcal{X}$ is a poor estimator in case of overfitting

$$E_{emp} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathcal{M}(\boldsymbol{x}^{(i)}) - \mathcal{M}^{\mathsf{PC}}(\boldsymbol{x}^{(i)})\right)^2$$

Leave-one-out cross validation

- From statistical learning theory, model validation shall be carried out using independent data

$$E_{LOO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathcal{M}(\boldsymbol{x}^{(i)}) - \mathcal{M}^{PC}(\boldsymbol{x}^{(i)})}{1-h_i}\right)^2$$

where h_i is the i-th diagonal term of matrix $\mathbf{A}(\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}$

Outline

Models with time-dependent outputs

Problem statement

- Consider a computational model of a dynamical system:

$$\mathcal{D}_{\boldsymbol{\Xi}} \times [0, T] : \ (\boldsymbol{\xi}, t) \mapsto \mathcal{M}(\boldsymbol{\xi}, t)$$

where $\boldsymbol{\Xi}$ is a random vector of uncertain parameters with given PDF $f_{\boldsymbol{\Xi}}$

- Uncertainties may be in:
  + The excitation, denoted by $x(\boldsymbol{\xi}_x, t)$
  + And/or in the system's characteristics $(\boldsymbol{\xi}_s)$:

i.e.:

$$\mathcal{M}(\boldsymbol{\xi}, t) \equiv \mathcal{M}(x(\boldsymbol{\xi}_x, t), \ \boldsymbol{\xi}_s)$$

PCEs for time-dependent outputs

Problem statement

$$\mathcal{M}^{\mathrm{PCE}}(\boldsymbol{\xi}, t) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}}(t)\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$$

Naive idea: time-frozen PCE

- Select an experimental design $\mathcal{E} = \left\{\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(n)}\right\}$, evaluate input excitation (if any), run the simulator and get a set of trajectories $\left\{\mathcal{M}(\boldsymbol{\xi}^{(i)}, t),\ i = 1, \ldots, n\right\}$
- By freezing time at a given $t_0 \in [0, T]$ one gets:

$$\mathcal{M}^{\mathrm{PCE}}(\boldsymbol{\xi}, t_0) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}}(t_0)\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$$

- Coefficients $\{y_{\boldsymbol{\alpha}}(t_0),\ \boldsymbol{\alpha} \in \mathcal{A}\}$ may be computed by standard techniques

Example: Duffing oscillator

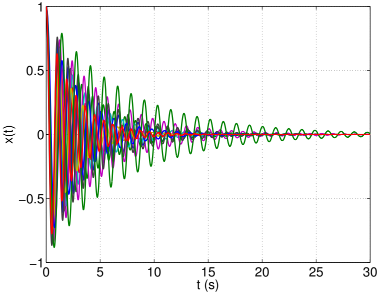
Non-linear SDOF Duffing oscillator:

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

Initial conditions: $x(0) = 1, \quad \dot{x}(0) = 0$

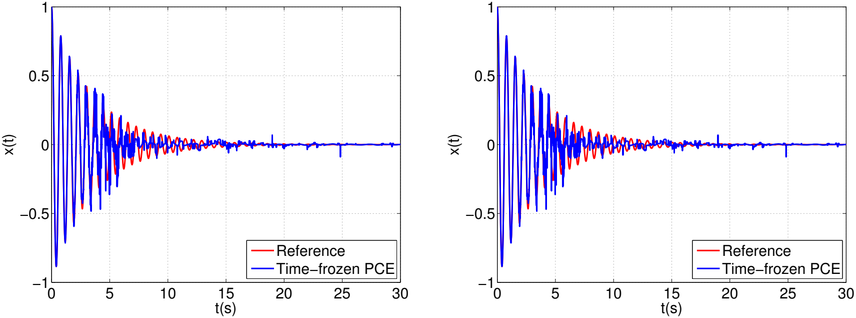
Input: 3 uniform random variables

RV	Distribution	Values
ζ	Uniform	$\mathcal{U}[0.015,\, 0.045]$
ω	Uniform	$\mathcal{U}[\pi,\, 3\,\pi]$
ε	Uniform	$\mathcal{U}[-0.25,\, -0.75]$



Samples of trajectories

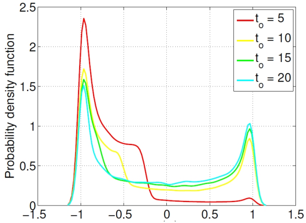
Time-frozen PCE

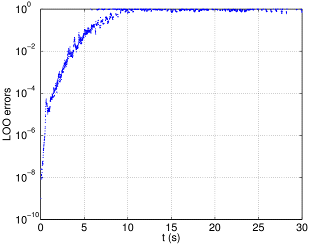


$(\zeta, \omega, \varepsilon) = (0.03, 8.92, -0.34)$ $(\zeta, \omega, \varepsilon) = (0.04, 3.18, -0.33)$**)**

Why time-frozen PCE does not work?

- The map $\boldsymbol{\xi} \mapsto \mathcal{M}(\boldsymbol{\xi}, t)$ becomes increasingly non linear with time
- The time-frozen distribution of the output at time t_0 becomes more complex (*e.g.* multimodal)
- Expansions of higher degree would be required to keep sufficient accuracy with time
- For a fixed experimental design, the LOO error blows up





Some literature

- Multi-elements PCEs: decomposition of the random space into non-overlapping sub-elements — Wan & Karniadakis, 2005
- Constant phase interpolation: responses interpolated in the phase space — Witteveen & Bijl, 2008
- Asynchronous time integration: intrusive transformed time variable introduced to reduce variability — Le Maître et al., 2010
- Time-dependent PCEs: new random variables added on-the-fly — Gerritsma et al., 2010
- PC flow map composition: long-term response obtained by composing intermediate PCE-based flow maps — Luchtenburg et al., 2014
- PC-NARX: future state determined by current and past states — Spiridonakos & Chatzi, 2015

Outline

Stochastic time warping

Heuristics

Le Maître *et al.* (2010)

Introduce a virtual time scale τ_i for each sample trajectory so that $y(\boldsymbol{\xi}^{(i)}, \tau_i)$ becomes "similar" to a reference trajectory

Measure of dissimilarity

$$\text{diss}\left[y(t)\,,\, y_{ref}(t)\right] \stackrel{\text{def}}{=} \frac{\left|\int_0^T y(t)\, y_{ref}(t)\, dt\right|}{\sqrt{\int_0^T y^2(t)\, dt \cdot \int_0^T y_{ref}^2(t)\, dt}}$$

- It is the cross-correlation of the two signals
- Bounded between 0 and 1

Stochastic time warping: procedure

Mai & Sudret (2015; 2016);

- Choose a reference trajectory $y_{ref}(t) = \mathcal{M}(\boldsymbol{\xi}_{ref},\, t)$ where *e.g.* $\boldsymbol{\xi}_{ref} = \mu_{\boldsymbol{\Xi}}$
- Define a stochastic time transform:

$$\tau(\boldsymbol{\xi}, t) = \mathcal{F}(\boldsymbol{\xi}, t) \quad \text{e.g. } \tau(\boldsymbol{\xi}, t) = \sum_{i=1}^{N_\tau} c_i(\boldsymbol{\xi_i})\, f_i(t)$$

  In practice: linear transform

$$\tau(\boldsymbol{\xi}) = k(\boldsymbol{\xi})\, t + \phi(\boldsymbol{\xi})$$

- For each sample trajectory $\{y_i(t),\, i = 1,\, \dots\, , n\}$, compute the appropriate rescaling:

$$(k_i,\, \phi_i) = \arg\min_{k,\phi} \text{diss}\left[\, y_i(k\, t + \phi),\, y_{ref}(t)\right]$$

- Compute a sparse PCE of the parameters of the time transform, *e.g.* :

$$k(\boldsymbol{\Xi}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} k_{\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\Xi}) \qquad \phi(\boldsymbol{\Xi}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \phi_{\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\Xi})$$

Stochastic time warping: procedure

- In the virtual time scale, trajectories show much higher coherency. τ-frozen PCE expansions apply:

$$y(\boldsymbol{\Xi}, \tau) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}}(\tau)\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\Xi})$$

Predictions for a new sample $\boldsymbol{\xi}^{(0)}$

- Predict the trajectory in the virtual time scale

$$y(\boldsymbol{\xi}^{(0)}, \tau) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}}(\tau)\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}^{(0)})$$

- Predict the proper time warping for this new trajectory:

$$\tau(\boldsymbol{\xi}^{(0)}) = k(\boldsymbol{\xi}^{(0)})\, t + \phi(\boldsymbol{\xi}^{(0)})$$

- Map back the predicted trajectory in the real time scale:

$$y(\boldsymbol{\xi}^{(0)}, t) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \left(k(\boldsymbol{\xi}^{(0)})\, t + \phi(\boldsymbol{\xi}^{(0)}) \right)\; \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}^{(0)})$$

Oregonator model

The Oregonator model represents a well-stirred, homogeneous chemical system governed by a three species coupled mechanism Le Maître et al. (2010)

Governing equations

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

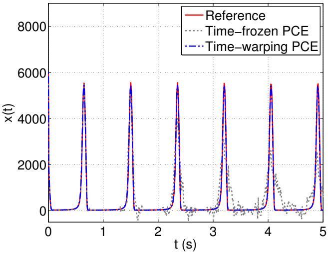
Input reaction parameters

Parameter	Distribution	Values
k_1	Uniform	$\mathcal{U}[1.8,\ 2.2]$
k_2	Uniform	$\mathcal{U}[0.095,\ 0.1005]$
k_3	Gaussian	$\mathcal{N}(104, 1.04)$
k_4	Uniform	$\mathcal{U}[0.0076, 0.0084]$
k_5	Uniform	$\mathcal{U}[23.4, 28.6]$

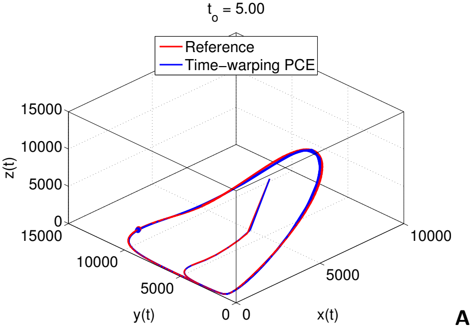
Oregonator model: prediction

Surrogate model

- Experimental design of size $n = 50$
- Validation set of size $n_{val} = 10,000$

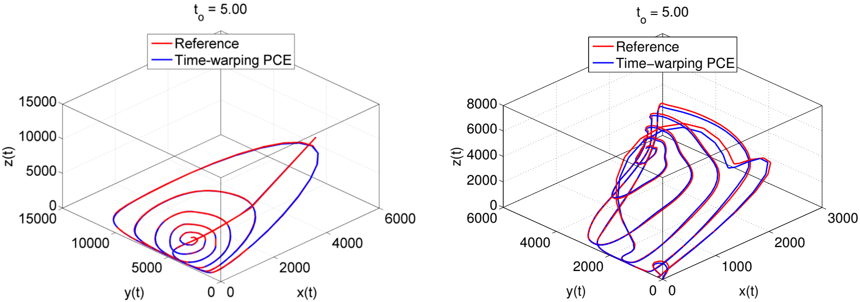


A specific trajectory ($\varepsilon = 0.0294$)



trajectory in the state-space

Oregonator model: mean and std trajectories



Mean trajectory ($\varepsilon \approx 10^{-4}$) **Standard deviation ($\varepsilon \approx 10^{-3}$)**

Bouc-Wen nonlinear oscillator

Governing equations

$$\ddot{y}(t) + 2\,\zeta\,\omega\,\dot{y}(t) + \omega^2(\rho\,y(t) + (1-\rho)\,z(t)) = -x(t)$$

$$\dot{z}(t) = \gamma\dot{y}(t) - \alpha\,|\dot{y}(t)|\,|z(t)|^{n-1}\,z(t) - \beta\,\dot{y}(t)\,|z(t)|^{n}$$

$$x(t) = A\,\sin(\omega_x\,t)$$

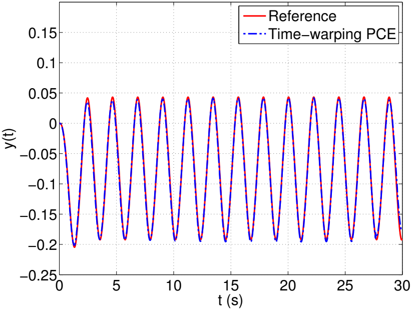
Input parameters

Parameter	Distribution	Mean	Standard deviation	COV
ζ	Uniform	0.02	0.002	0.1
ω	Uniform	$2\,\pi$	$0.2\,\pi$	0.1
α	Uniform	50	5	0.1
A	Uniform	1	0.1	0.1
ω_x	Uniform	π	$0.1\,\pi$	0.1

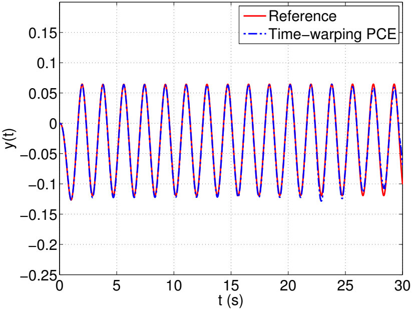
Bouc-Wen model: two particular predictions

Surrogate model

- Experimental design of size $n = 100$
- Validation set of size $n_{val} = 10,000$

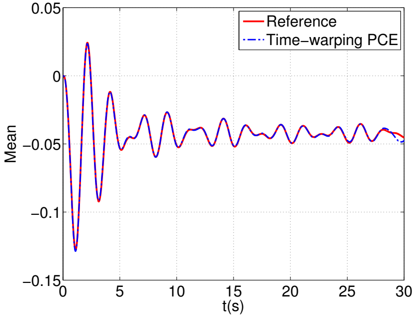


Trajectory #1 ($\varepsilon = 3.1 \cdot 10^{-3}$)

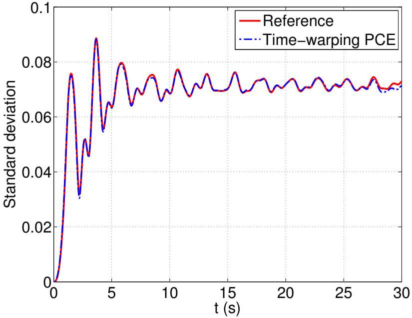


Trajectory #2 ($\varepsilon = 3.9 \cdot 10^{-3}$)

Bouc-Wen model: statistical moments



Mean trajectory ($\varepsilon = 2.4 \cdot 10^{-3}$)

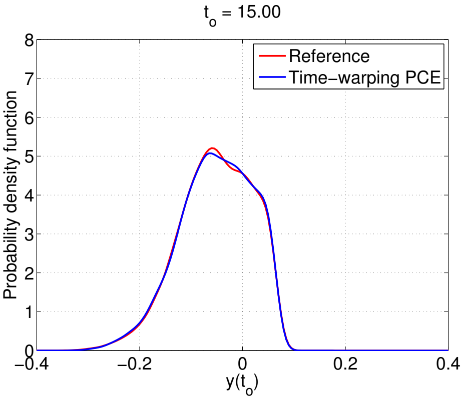


Standard deviation ($\varepsilon = 2.4 \cdot 10^{-3}$)

Bouc-Wen model: evolution of PDF

Surrogate model

- Experimental design of size $n = 100$
- Validation set of size $n_{val} = 10,000$



Time-warping PCEs capture not only the mean and standard deviation but also the entire PDF

Outline

Nonlinear AutoRegressive with eXogenous input model

NARX model

Billings, 2013

Based on a time-dependent input excitation $x(t)$ and corresponding system response $y(t)$, the dynamics is captured through:

$$y(t) = \mathcal{F}\left(x(t), \ldots, x(t-n_x), y(t-1), \ldots, y(t-n_y)\right) + \varepsilon_t$$

where:

- $\boldsymbol{z}(t) = \left(x(t), \ldots, x(t-n_x), y(t-1), \ldots, y(t-n_y)\right)^{\mathsf{T}}$ is the vector of current and past values
- n_x and n_y denote the maximum input and output time lags
- $\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2(t))$ is the residual error
- $\mathcal{F}(\cdot)$ is a functional of NARX terms, usually linear-in-parameters:

$$y(t) = \sum_{i=1}^{n_g} \vartheta_i \, g_i(\boldsymbol{z}(t)) + \varepsilon_t$$

PC-NARX model

Spiridonakos *et al.* , 2015a,2015b

Computational model with uncertainties

$$y(t, \boldsymbol{\xi}_x, \boldsymbol{\xi}_s) \stackrel{\text{def}}{=} \mathcal{M}(x(t, \boldsymbol{\xi}_x), \boldsymbol{\xi}_s)$$

- $\boldsymbol{\xi}_x$: uncertainty in the input excitation
- $\boldsymbol{\xi}_s$: uncertainty in the system

PC-NARX expansion

$$y(t, \boldsymbol{\xi}) = \sum_{i=1}^{n_g} \vartheta_i(\boldsymbol{\xi})\, g_i(\boldsymbol{z}(t)) + \varepsilon_g(t, \boldsymbol{\xi}) \qquad \boldsymbol{\xi} = (\boldsymbol{\xi}_x, \boldsymbol{\xi}_s)$$

The NARX stochastic coefficients $\vartheta_i(\boldsymbol{\xi})$ are represented by PCEs:

$$\vartheta_i(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_i} \vartheta_{i,\boldsymbol{\alpha}}\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$$

PC-NARX model

$$y(t,\boldsymbol{\xi}) = \sum_{i=1}^{n_g} \sum_{\boldsymbol{\alpha}\in\mathcal{A}_i} \vartheta_{i,\boldsymbol{\alpha}}\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\, g_i(\boldsymbol{z}(t)) + \varepsilon(t,\boldsymbol{\xi})$$

Interpretation

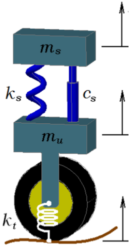
- PC-NARX is a NARX model in which each (random) coefficient is expanded as a PCE
- Compared to time-frozen PCE, a specific dynamics of the random coefficients is imposed
- Similar to flow map composition since the response at current instant is used to predict the response at future instants

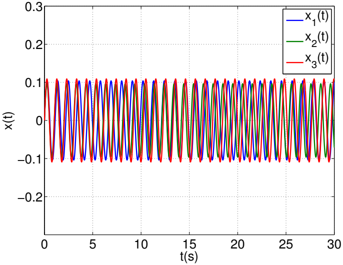
Experimental design

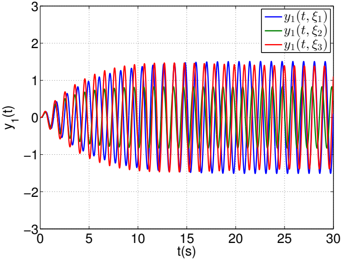
Data

- N realizations of the input excitation, cast as $(x_k[1], \ldots, x_k[T])^\mathsf{T}$, $k = 1, \ldots, N$ (T time instants)
- The corresponding system response computed by a simulator, cast as $(y_k[1], \ldots, y_k[T])^\mathsf{T}$

Example: quarter car model







Deterministic NARX calibration

For a particular realization $\boldsymbol{\xi}_k$

- Select NARX model (candidate terms):

$$\boldsymbol{z}(t) = (x(t), \ldots, x(t-n_x), y(t-1), \ldots, y(t-n_y))^{\mathsf{T}}$$
$$\boldsymbol{\phi}(t) = \{g_i(\boldsymbol{z}(t)), \, i = 1, \ldots, n_g\}^{\mathsf{T}}$$

- Use least angle regression (LARS) to select the best explanatory subset of terms Efron *et al.* , 2004

- Compute the coefficients $\boldsymbol{\vartheta}_k$ by ordinary least-squares

Prediction error (of model $\#k$ on trajectory l)

$$\varepsilon_l^{\#k} = \frac{\sum_{t=1}^{T} (y(t, \boldsymbol{\xi}_l) - \hat{y}^{\#k}(t, \boldsymbol{\xi}_l))^2}{\sum_{t=1}^{T} (y(t, \boldsymbol{\xi}_l) - \bar{y}(t, \boldsymbol{\xi}_l))^2}$$

Common NARX basis

Premise

To expand the NARX coefficients onto a PC basis, it is necessary to have a common NARX model for all trajectories

Procedure

- Select $K \leq N$ trajectories ("NARX learning set"), *e.g.* with the strongest non linear behaviour (peak displacement, velocities, etc.)
- Determine the sparse deterministic NARX models for realizations $k = 1, \ldots, K$, which leads to $P \leq K$ different possible models called $\#1, \ldots, \#P$
- Compute the NARX coefficients of the N trajectories, for each model $\#p$, and evaluate an average error:

$$\varepsilon_p = \frac{1}{N} \sum_{k=1}^{N} \varepsilon_k^{\#p}$$

- Select the final best NARX model that minimizes ε_p

PCE of the NARX coefficients

PCE calibration

- Once a common NARX basis has been found, N realizations of the NARX coefficients are available:

$$\mathcal{E}\mathcal{D} = \{\vartheta_{i,k}, i = 1, \ldots, n_g;\ k = 1, \ldots, N\}$$

- n_g different sparse PC expansions are built from this experimental design, using least-angle regression (LAR) Blatman & Sudret, 2011

$$\vartheta_i(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_i} \vartheta_{i,\boldsymbol{\alpha}}\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$$

PC-NARX prediction

- For a new realization of the input parameters $\boldsymbol{\xi}_0$, the NARX coefficients are first evaluated from PCEs
- Then they are plugged into the NARX model

Bouc-Wen model

Governing equations

Kafali & Grigoriu (2007), Spiridonakos & Chatzi (2015)

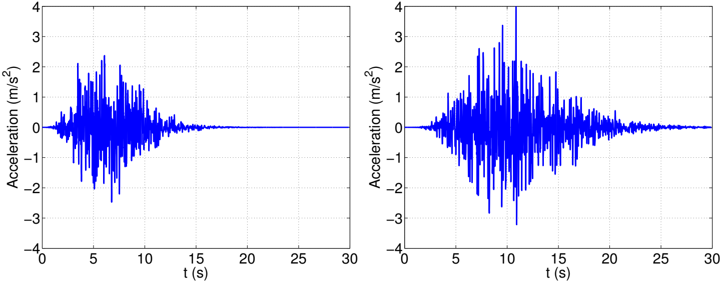
$$\ddot{y}(t) + 2\,\zeta\,\omega\,\dot{y}(t) + \omega^2(\rho\,y(t) + (1-\rho)\,z(t)) = -x(t),$$
$$\dot{z}(t) = \gamma\dot{y}(t) - \alpha\,|\dot{y}(t)|\,|z(t)|^{n-1}\,z(t) - \beta\,\dot{y}(t)\,|z(t)|^n\,,$$

Excitation

$x(t)$ is generated by a probabilistic ground motion model

Rezaeian & Der Kiureghian (2010)

$$x(t) = q(t, \boldsymbol{\alpha}) \sum_{i=1}^{n} s_i\left(t, \boldsymbol{\lambda}(t_i)\right)\, U_i$$

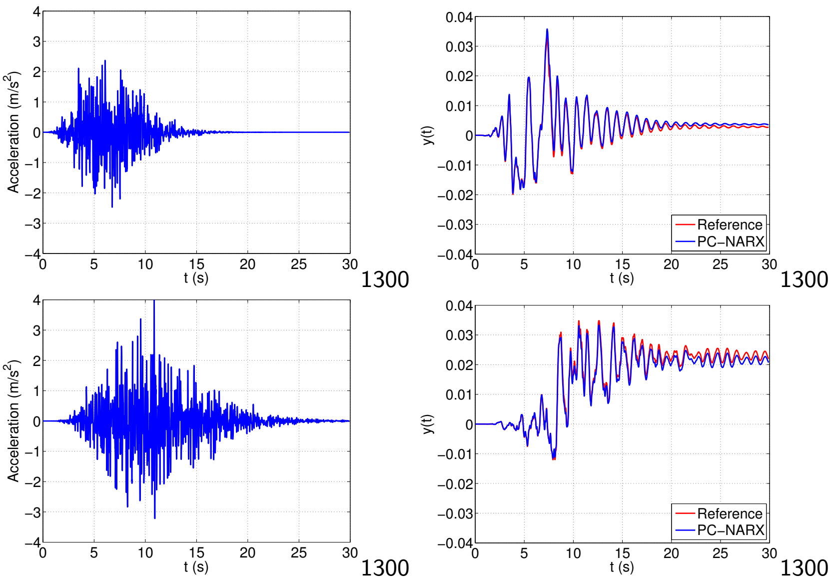


Bouc-Wen model

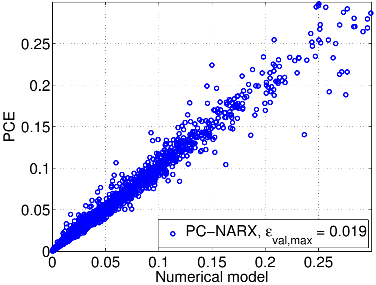
Marginal distributions of the model parameters

Parameters	Distribution	Support	Mean	Std
ω (rad/s)	Uniform	$[5.373, 6.567]$	5.97	0.3447
α (1/m)	Uniform	$[45, 55]$	50	2.887
I_a (s.g)	Lognormal	$(0, +\infty)$	0.0468	0.164
D_{5-95} (s)	Beta	[5, 45]	17.3	9.31
t_{mid} (s)	Beta	[0.5, 40]	12.4	7.44
$\omega_{mid}/2\pi$ (Hz)	Gamma	$(0, +\infty)$	5.87	3.11
$\omega'/2\pi$ (Hz)	Two-sided exponential	[-2, 0.5]	-0.089	0.185
ζ_f (.)	Beta	[0.02, 1]	0.213	0.143

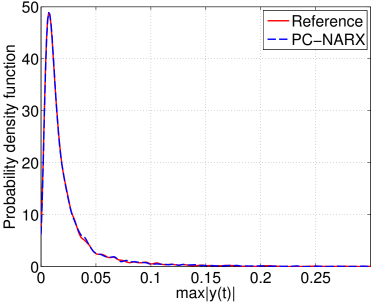
Bouc-Wen model: prediction



Bouc-Wen model: prediction



Maximal displacements



PDF of maximal displacements

Conclusions

- Surrogate models are unavoidable for solving uncertainty quantification problems involving costly computational models (*e.g.* transient finite element models)
- For uncertain dynamical systems under uncertain excitation, time-frozen PCE usually does not work
- Proper pre-processing using time warping or NARX modelling allows to transform the data into an auxiliary space suitable for PC expansions
- Extensions to space-time variant problems are currently investigated

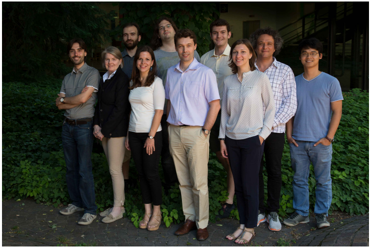
References

Mai, C. and B. Sudret (2016).
Surrogate models for oscillatory systems using sparse polynomial chaos expansions and stochastic time warping.
Int. J. Uncertainty Quantification.
Submitted.

Mai, C. V., M. Spiridonakos, E. N. Chatzi, and B. Sudret (2016).
Surrogate modelling for stochastic dynamical systems by combining NARX models and polynomial chaos expansions.
Int. J. Uncertainty Quantification.
Accepted.

Yaghoubi, V., S. Marelli, B. Sudret, and T. Abrahamsson (2016).
Sparse polynomial chaos expansions of frequency response functions using stochastic frequency transformation.
Prob. Eng. Mech.
Submitted.

Questions ?



Chair of Risk, Safety & Uncertainty Quantification

www.rsuq.ethz.ch

The Uncertainty Quantification Laboratory

www.uqlab.com

Thank you very much for your attention !