# A feature mapping strategy of metamodelling for nonlinear stochastic dynamical systems with low to high-dimensional input uncertainties

Zhiqiang Wan[a], Jianbing Chen[b,c], Weifeng Tao[a,*], Pengfei Wei[d], Michael Beer[e,f,c], Zhongming Jiang[g]

[a]*School of Mechanics, Civil Engineering and Architecture, Northwestern Polytechnical University, Xi'an 710072, P. R. China*
[b]*State Key Laboratory of Disaster Reduction in Civil Engineering & College of Civil Engineering, Tongji University, Shanghai 200092, P. R. China*
[c]*International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai 200092, P. R. China*
[d]*School of Power and Energy, Northwestern Polytechnical University, Xi'an 710072, P. R. China*
[e]*Institute for Risk and Reliability, Leibniz Universität Hannover, Hannover 30167, Germany*
[f]*Institute of Risk and Uncertainty, University of Liverpool, Liverpool L69 7ZF, United Kingdom*
[g]*College of Engineering and Technology, Southwest University, Chongqing 400700, P. R. China*

## Abstract

This paper deals with the issue on metamodelling (a.k.a. surrogate modelling) of nonlinear stochastic dynamical systems, which are often with multiple input uncertainties $\boldsymbol{\Theta} \in \mathbb{R}^n$, viz., the dimension $n$ may range from low to high (e.g., $n \geq 10$). In this paper, to circumvent the problem of "curse of dimensionality" of high-dimensional input uncertainties, the feature spaces of outputs and inputs are firstly extracted from the original output and input spaces, and thus a feature mapping strategy is proposed. To form the feature output space, the nonlinear autoregressive with exogenous inputs (NARX) and the proper orthogonal decomposition (POD) are adopted, while the feature input space is detected by the active subspace method (ASM). It is found that the dimension of feature input (output) space may be much less than the one of original input (output) space, thus the applicability of many metamodelling methods can be naturally enhanced. On the constructed input-output feature space, the procedure of metamodelling is completed by the polynomial chaos expansion (PCE) combined with Kriging, which can capture global behaviours as well as local characteristics of the computational model. Two techniques are introduced to accelerate the proposed feature mapping strategy, consisting of the GF-discrepancy minimization algorithm for the design of experiments (DoEs), and the manifold optimization technique for the parameter identification of ASM. Four benchmarks, including a mathematical function ($n = 2$), a dynamical quarter car model ($n = 10$), a Bouc-Wen nonlinear oscillator subjected to earthquake ground motions ($n = 30$), and the first sub-system (as a black box) of the NASA UQ Challenge 2019 ($n = 100$), are studied to demonstrate the accuracy and efficiency of the proposed method. Some problems to be further studied are also outlined.

*Keywords:* metamodel, high-dimensionality, feature mapping, nonlinear autoregressive with exogenous inputs (NARX), proper orthogonal decomposition (POD), active subspace method (ASM), PC-Kriging

## Contents

[*]Corresponding author.

*Email addresses:* wanzhiqiang@nwpu.edu.cn (Zhiqiang Wan), chenjb@tongji.edu.cn (Jianbing Chen), wftao2048@nwpu.edu.cn (Weifeng Tao), pengfeiwei@nwpu.edu.cn (Pengfei Wei), beer@irz.uni-hannover.de (Michael Beer), jiangzm@swu.edu.cn (Zhongming Jiang)

## 1. Introduction

Modern structural analysis has been rapidly developed with the aid of advanced computers, and the numerical computation is gradually becoming one of the most critical parts of structural analysis and design [1]. Simultaneously, the cost of computation on dealing with large and complex structures, e.g., state-of-the-art finite element models, is increasing as well. In particular, when facing with issues involved with high-dimensional input uncertainties[1], the model evaluation may cost hours to days, which is prohibitively time-consuming for engineering practice. Instead, the utilization of *metamodels* (a.k.a. surrogate models) that only takes seconds or minutes, has shown advantages on handling with real engineering problems, such as reliability assessment [2, 3], sensitivity analysis [4] and structural optimization [5], etc.

During the past decades, a variety of metamodelling methods have been developed, such as response surface method (RSM) [6, 7], Kriging [3, 8, 9], polynomial chaos expansion (PCE) [10, 11, 12], support vector machine (SVM) [13] and artificial neutral network (ANN) [14, 15, 16], etc. In particular, the metamodel has become one of the powerful tools for analysis in the field of uncertainty quantification (UQ) that increasingly attracts much attention of researchers and engineering pioneers [17]. A comprehensive literature review of widely used metamodels can be found in [18], which provides a practical guidance on selecting different metamodels by an integrated measure of computational time, model accuracy and problem size. Besides, comparative studies of metamodels including RSM, Kriging, PCE and SVM for reliability analysis and reliability-based design optimization (RBDO) are well discussed in [19] and [5], and state-of-the-art literature reviews of Kriging and PCE with fruitful numerical illustrations are available in [20] and [21], respectively.

Most metamodels have been verified to be accurate and efficient for predicting model evaluations with relatively low-dimensional inputs to low-dimensional outputs. However, it has been recognized that for the case of high-dimensionality, most metamodels are haunted by the issue of "curse of dimensionality" [22], which makes the construction of a reliable metamodel become distinctly costly. Moreover, under the high-dimensional circumstance, there may be ancillary model errors due to the redundant complexity of metamodels [23].

To this end, it is natural to ensure and improve the applicability of metamodels by adopting *dimension reduction* approaches, such as high dimensional model representation (HDMR) [24], sensitivity analysis (SA) [25], principal component analysis (PCA) [26] and sparse PCE [27], etc., which are generally suitable for static systems. The HDMR decomposes the original model of high-dimensional inputs into a summation of a series of independent models involved with single or interacted inputs, and it has been known that usually only few interactions of higher-orders are critical thus a truncation can be done [28]. The same thought appears in the SA, where it states that the model outputs may be mainly dependent on several

[1] In this paper, the input uncertainties are to refer in particular to the random variables (RVs) of the system input. Besides, without confusion, the "high-dimensional input uncertainties" is defined by that if the number of RVs is more than 10.

dominant input variables [29]. Similarly, the sparse PCE decreases the number of PC terms by removing non-significant coefficients. From the perspective of the PCA it reveals that there may exist some influential directions or feature planes in high-dimensional dataset. Based on the above heuristic ideas, the active subspace method (ASM) [30] firstly generates the covariance matrix of gradients of quantity of interest (QoI), and then detects the *active subspace* whose dimension may be far less than that of the original input space. Towards the dimension reduction of dynamical systems, there are typical methods based on nonlinear autoregressive with exogenous inputs (NARX) [31], proper orthogonal decomposition (POD) [32] and others [33]. In general, NARX can be flexibly enhanced by well understanding the physical mechanisms of the studied systems [33], while POD is much more universal since it only relies on the quantity and quality of the given dataset.

To the best of the authors' knowledge, in spite of copious amounts of dimension reduction methods and metamodels that have been developed, there is no much attention on metamodelling strategies based on the framework of feature mappings between the dimension-reduced input space and the dimension-reduced output space, especially for the issue on metamodelling of high-dimensional and nonlinear stochastic dynamical systems. On this basis, by taking advantages of NARX, POD and ASM, a feature mapping strategy for metamodelling is proposed in the present paper. The NARX and POD methods are employed to represent the feature output space that extracted from the high-dimensional dynamical responses of interest, while the ASM is adopted to build the feature input space that detected from the high-dimensional input variables. Furthermore, to reduce metamodelling errors, the PC-Kriging metamodel [34] is adopted and constructed with the aid of the GF-discrepancy minimization technique [35] and the Grassmann manifold optimization algorithm [36].

The remainder of the present paper is organized as follows. In Section 2, the NARX, POD and ASM are primarily summarized. Whereafter, the proposed feature mapping strategy is elaborated in Section 3. The numerical scheme consisting of the GF-discrepancy minimization technique, the manifold optimization algorithm and the PC-Kriging metamodel will be introduced in this section as well. In Section 4, four numerical examples are studied to demonstrate the accuracy and efficiency of the proposed method in detail. Conclusions and some other aspects to be further studied are summarized in Section 5.

## 2. Dimension reduction methods

In this section, the basic ideas of three dimension reduction methods, i.e., the nonlinear autoregressive with exogenous input (NARX) model, the proper orthogonal decomposition (POD) and the active subspace method (ASM) will be briefly outlined. These methods will be the bricks of the proposed approach.

Without loss of generality, consider a stochastic dynamical system in the form of state space:

$$\dot{\boldsymbol{X}} = \mathbf{G}\left(\boldsymbol{X}, \boldsymbol{\Theta}, t\right), \ \ \boldsymbol{X}(t_0) = \boldsymbol{x}_0 \tag{1}$$

where $\mathbf{G}(\cdot)$ stands for a linear or nonlinear state mapping, $\boldsymbol{X} = (X_1, X_2, \cdots, X_s)^\top$ denotes an $s$-dimensional state vector and its velocity is defined by $\dot{\boldsymbol{X}} = \partial \boldsymbol{X}/\partial t$. The input $n$-dimensional random vector is denoted by $\boldsymbol{\Theta} = (\Theta_1, \Theta_2, \cdots, \Theta_n)^\top$ with a known joint probability density function (PDF) $p_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$. In general, $\boldsymbol{\Theta}$ is a combined vector of $\boldsymbol{\Theta}_{\mathrm{E}}$ and $\boldsymbol{\Theta}_{\mathrm{S}}$, where $\boldsymbol{\Theta}_{\mathrm{E}}$ results from the decomposition of the stochastic excitation $\xi(t)$, e.g., by Karhunen–Loève decomposition [37], while $\boldsymbol{\Theta}_{\mathrm{S}}$ represents the randomness of basic mechanical parameters of the system.

For most well-posed engineering problems, the solution of Eq. (1) exists and uniquely depends on the initial conditions $\boldsymbol{x}_0$ and input parameters $\boldsymbol{\theta}$ (a realization of $\boldsymbol{\Theta}$). In this sense, the QoI as a component of solution of Eq. (1) can be rewritten by

$$X(t) = \mathcal{G}\left(\xi; \boldsymbol{\Theta}\right) \tag{2}$$

where the stochastic process $\xi$ is a function of $\boldsymbol{\Theta}_{\mathrm{E}}$ that omitted in the formula for the sake of simplicity, and here $s = 1$ is considered for reasons of simplicity as well. Obviously, Eq. (2) is a stochastic parametric system and the QoI $X(t)$ is a stochastic process.

In the numerical implementation, the solution of Eq. (2) is usually limited to a concerned time interval $[0, T]$, and thus for each determined realization $\boldsymbol{\theta}$ the solution $x(t)$ is numerically discretized into $N_t$ time steps by the index $t = 1, 2, \cdots, N_t$ with respect to (w.r.t.) the real time duration $0, \Delta t, 2\Delta t, \cdots, (N_t - 1)\Delta t = T$. Therefore, the discretized $x(t)$ can be regarded as an $N_t$-dimensional vector $\boldsymbol{x}$, such that

$$\boxed{x(t) := \boldsymbol{x} = \mathcal{G}\left(\boldsymbol{\xi}; \boldsymbol{\theta}\right), \ \boldsymbol{\theta} \in \mathbb{R}^n \mapsto \boldsymbol{x} \in \mathbb{R}^{N_t}} \tag{3}$$

where $\boldsymbol{\xi}$ is an $N_t$-dimensional vector w.r.t. the stochastic process $\xi(t)$ as well.

### *2.1. Nonlinear autoregressive with exogenous input (NARX)*

The NARX model provides an approach to reconstruct Eq. (3) at specific time instants as the linear combinations of $n_g$ linear or nonlinear functions of its past values and values of past and current instants of the exogenous input [31]:

$$x(t) := \sum_{i=1}^{n_g} \vartheta_i g_i(y(t)) + \epsilon(t), \ t = 1, 2, \cdots, N_t \tag{4}$$

where the $i$-th NARX term $g_i(\cdot)$ is a linear or nonlinear function of

$$y(t) \subseteq \{x(t-1), x(t-2), \cdots, x(t-n_x), \xi(t), \xi(t-1), \cdots, \xi(t-n_\xi)\} \tag{5}$$

and $\vartheta_i$ is the corresponding $i$-th NARX coefficient, $n_x$ and $n_\xi$ are time lags of the processes $x(t)$ and $\xi(t)$ respectively, and $\epsilon(t)$ is the residual error.

The time lags $n_x$ and $n_\xi$, as well as the specific forms of functions $g_i(\cdot)$ for $i = 1, 2, \cdots, n_g$, are all prescribed. When the physical mechanisms of the system are well-known, it is relatively easy to determine these undetermined parameters and forms of functions. However, for most cases, the studied system is considered to be a "black box". To this end, a model selection strategy is of need, and as suggested in [33] it mainly consists of 2 steps:

**Step 1.** Generate $N_g$ "candidate" NARX terms. For instance, the constant, the polynomial types $x^{l_x}(t-n_x)\xi^{l_\xi}(t-n_\xi)$, or the trigonometric types $\sin^{l_x}(x(t-n_x))\cos^{l_\xi}(\xi(t-n_\xi))$, etc., for $n_x = 1, 2, \cdots$, $n_\xi = 0, 1, \cdots$ and $l_x, l_\xi = 0, 1, \cdots$. Although it is still somehow subjective to determine these candidate terms, each term can be regarded as a sub-model that represents a partial property of the dynamical responses. In this sense, it provides a potential development of the method by introducing more physical mechanisms.

**Step 2.** Select most relevant NARX terms ($1 \leq n_g \leq N_g$) under a given tolerance $\epsilon_{\text{tol}}$, e.g., $10^{-5} \sim 10^{-6}$. In this paper, the least angle regression (LARS) algorithm [38] is employed as suggested in [33].

From Eqs. (4) and (5), it notes that $x(t)$ and $y(t)$ are stochastic processes w.r.t. the random vector $\mathbf{\Theta}$. Suppose we have $N$ samples $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_N\} \sim p_{\mathbf{\Theta}}(\boldsymbol{\theta})$ with corresponding solutions $\boldsymbol{x}_k = \mathcal{G}(\boldsymbol{\xi}_k, \boldsymbol{\theta}_k)$ for $k = 1, 2, \cdots, N$. For the $k$-th sample, denote $\phi_k(t) = [g_1(y(t); \boldsymbol{\theta}_k), g_2(y(t); \boldsymbol{\theta}_k), \cdots, g_{n_g}(y(t); \boldsymbol{\theta}_k)]^\top \in \mathbb{R}^{n_g}$ as the NARX term vector at the time instant $t$, and define the information matrix $\mathbf{\Phi}_k$ by

$$\mathbf{\Phi}_k = \begin{bmatrix} \phi_k(1)^\top \\ \vdots \\ \phi_k(N_t)^\top \end{bmatrix} \in \mathbb{R}^{N_t \times n_g} \tag{6}$$

Then Eq. (4) can be rewritten in a matrix form as

$$\boldsymbol{x}_k = \mathbf{\Phi}_k \boldsymbol{\vartheta}_k + \boldsymbol{\epsilon}_k, \ \ k = 1, 2, \cdots, N \tag{7}$$

where $\boldsymbol{\vartheta}_k = [\vartheta_1(\boldsymbol{\theta}_k), \vartheta_2(\boldsymbol{\theta}_k), \cdots, \vartheta_{n_g}(\boldsymbol{\theta}_k)]^\top \in \mathbb{R}^{n_g}$ is the NARX coefficient vector of the $k$-th sample, and $\boldsymbol{\epsilon}_k \in \mathbb{R}^{N_t}$ is the $k$-th residual error. Then the NARX coefficient vector can be solved by minimizing the mean-squared error of $\boldsymbol{\epsilon}_k$ in Eq. (7), which leads to the ordinary least squares (OLS) solution:

$$\boldsymbol{\vartheta}_k = \arg\min_{\boldsymbol{\vartheta}} \left(\boldsymbol{\epsilon}_k^\top \boldsymbol{\epsilon}_k\right) = \left(\mathbf{\Phi}_k^\top \mathbf{\Phi}_k\right)^{-1} \mathbf{\Phi}_k^\top \boldsymbol{x}_k, \ \ k = 1, 2, \cdots, N \tag{8}$$

It should be emphasized that for different samples it may adaptively select different NARX terms, and possibly the number of NARX terms $n_g$ may also be different. This is exactly why stochastic systems greatly differ from deterministic systems. In this sense, an optional strategy is to pick some most "influential" samples whose absolute maximum response exceeds a given large value [33]. Specifically, this large number denoted by $x_{\text{lim}}$ can be determined as the threshold in the reliability analysis. Then the problem of model selection in Eq. (8) can be redefined by

$$\boldsymbol{\vartheta}_k = \arg\min_{\vartheta} \left(\boldsymbol{\epsilon}_k^\top \boldsymbol{\epsilon}_k\right) = \left(\mathbf{\Phi}_k^\top \mathbf{\Phi}_k\right)^{-1} \mathbf{\Phi}_k^\top \boldsymbol{x}_k, \ \ k = 1, 2, \cdots, K \tag{9}$$

which satisfies that $\max|\boldsymbol{x}_k| \geq x_{\mathrm{lim}}$ and $K < N$. Moreover, the relative error of prediction of the $k$-th NARX model on the $j$-th sample can be defined by

$$\epsilon_j^{(k)} = \frac{\left(\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_j^{(k)}\right)^\top \left(\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_j^{(k)}\right)}{\left(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_j\right)^\top \left(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_j\right)}, \ j = 1, 2, \cdots, N \tag{10}$$

where $\tilde{\boldsymbol{x}}_j^{(k)}$ is the $j$-th predicted response via the $k$-th NARX model, and $\bar{\boldsymbol{x}}_j$ is the mean response of the $j$-th sample. On this basis, the mean value of the relative errors in Eq. (10) can be taken as the error criterion for selecting the optimal NARX model in Eq. (9) with an acceptable error $\epsilon$, i.e.,

$$\bar{\epsilon}^{(k)} = \frac{1}{N}\sum_{j=1}^{N} \epsilon_j^{(k)} \Rightarrow \min(\bar{\epsilon}^{(1)}, \bar{\epsilon}^{(2)}, \cdots, \bar{\epsilon}^{(K)}) \leq \epsilon = 10^{-3} \tag{11}$$

Once the optimal NARX model is constructed, i.e., the NARX terms $\left\{g_1(\cdot), g_2(\cdot), \cdots, g_{n_g}(\cdot)\right\}$ are determined, the NARX model representation of the QoI $\boldsymbol{x} \in \mathbb{R}^{N_t}$ in Eq. (7) is then formulated by

$$\boxed{\boldsymbol{x} \xrightarrow{\text{NARX}} \tilde{\boldsymbol{x}} = \boldsymbol{\Phi}\boldsymbol{\vartheta}} \tag{12}$$

where $\boldsymbol{\vartheta} \in \mathbb{R}^{n_g}$ and the NARX coefficients $\boldsymbol{\vartheta}_k$ for $k = 1, 2, \cdots, N$ is calculated via the least-squares solution in Eq. (8), respectively.

### *2.2. Proper orthogonal decomposition (POD)*

The proper orthogonal decomposition (POD) has been widely studied in various fields for decades [39], and has been named by Karhunen-Loève (KL) decomposition, principal component analysis (PCA), or singular value decomposition (SVD), etc., depending on specific fields of application. Generally speaking, POD is a method to extract the coherent modes from an ensemble of signals —the high-dimensional vector $\boldsymbol{x}_k \in \mathbb{R}^{N_t}$ for $k = 1, 2, \cdots, N$ as defined above. The decomposition procedure is conducted as follows:

**Step 1.** Denote the signal matrix $\boldsymbol{\mathcal{X}} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{N_t \times N}$, and calculate its SVD by

$$\boldsymbol{\mathcal{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \tag{13}$$

where $\boldsymbol{U} \in \mathbb{R}^{N_t \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times r}$ are orthogonal matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$ is a sparse matrix with only non-negative diagonal values denoted by $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ and $r = \min(N_t, N)$.

**Step 2.** Determine the smallest rank $n_t$ that satisfies

$$1 - \frac{\sum_{j=1}^{n_t} \sigma_j^2}{\sum_{i=1}^{r} \sigma_i^2} \leq \epsilon \text{ for } 1 \leq n_t \leq r \tag{14}$$

where $\epsilon > 0$ is a given small value, e.g., $\epsilon = 10^{-3}$ to make sure 99.9% energy of the signal is reserved [32]. Then, the POD approximation is generated by $\hat{\boldsymbol{\mathcal{X}}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^\top$ where $\hat{\boldsymbol{U}} \in \mathbb{R}^{N_t \times n_t}$ (the first $n_t$

columns of $\boldsymbol{U}$), $\hat{\boldsymbol{V}} \in \mathbb{R}^{N \times n_t}$ (the first $n_t$ columns of $\boldsymbol{V}$) and $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{n_t \times n_t}$ (the first $n_t$ columns and rows of $\boldsymbol{\Sigma}$). Therefore, due to the Eckart–Young–Mirsky theorem, the approximation error of POD is quantified by the Frobenius norm, i.e.,

$$\begin{aligned} \left\| \boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}} \right\|_{\mathrm{F}} &= \sqrt{\sigma_{n_t+1}^2 + \sigma_{n_t+2}^2 + \cdots + \sigma_r^2} \\ &\leq \epsilon^{1/2} \sqrt{(\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_{n_t}^2)} \end{aligned} \tag{15}$$

**Step 3.** On this basis, the dimension of the QoI $\boldsymbol{x} \in \mathbb{R}^{N_t}$ can be reduced from $N_t$ to $n_t$, i.e.,

$$\boxed{\boldsymbol{x} \xrightarrow{\text{POD}} \tilde{\boldsymbol{x}} = \hat{\boldsymbol{U}} \hat{\boldsymbol{x}}} \tag{16}$$

where $\hat{\boldsymbol{x}} \in \mathbb{R}^{n_t}$ is calculated by $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{U}}^\top \boldsymbol{x}_k$ for $k = 1, 2, \cdots, N$.

### *2.3. Active subspace method (ASM)*

Unlike the aforementioned NARX and POD that focus on the dimension reduction of the QoI, i.e., decreasing the dimensionality of $\boldsymbol{x}$, the active subspace method (ASM) is aimed at detecting the most variant dimensionality of inputs $\boldsymbol{\theta}$. The following recalls basic concepts of ASM in Refs. [30, 40, 41].

Consider a mapping $f : \boldsymbol{\theta} \in \mathbb{R}^n \mapsto x \in \mathbb{R}^1$ where the uncertain parameters $\boldsymbol{\theta}$ equipped with a triple $(\Omega, \mathcal{F}, \mathbb{P})$ has the joint PDF $p_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$. Assume $f$ is absolutely continuous and square-integrable w.r.t. $p_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$. Then, denote the uncentered covariance matrix $\mathbf{C}$ as

$$\mathbf{C} := \int_{\Omega} \left(\nabla_{\boldsymbol{\theta}} f\right) \left(\nabla_{\boldsymbol{\theta}} f\right)^\top p_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} = \mathbb{E}\left[\left(\nabla_{\boldsymbol{\Theta}} f\right) \left(\nabla_{\boldsymbol{\Theta}} f\right)^\top\right] \in \mathbb{R}^{n \times n} \tag{17}$$

where the column vector $\nabla_{\boldsymbol{\theta}} f = \left[\frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2}, \cdots, \frac{\partial f}{\partial \theta_n}\right]^\top \in \mathbb{R}^n$ is the gradient of $f$ and $\mathbb{E}[\cdot]$ is the expected operator. Then the eigenvalue decomposition of $\mathbf{C}$ yields

$$\mathbf{C} = \mathbf{W} \boldsymbol{\Lambda} \mathbf{W}^\top \tag{18}$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the eigenvector matrix and $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ is the eigenvalue matrix with only eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n \geq 0$ lying on its diagonal.

The active input dimension $m$ is defined by [30]

$$m := \underset{j \in \{1, 2, \cdots, n-1\}}{\arg\max} \left| \lambda_j - \lambda_{j+1} \right| \tag{19}$$

which shows the largest gap between two adjoint eigenvalues. Then, the original inputs $\boldsymbol{\theta} \in \mathbb{R}^n$ is transformed into the active inputs $\hat{\boldsymbol{\theta}} \in \mathbb{R}^m$ by

$$\boxed{\boldsymbol{\theta} \xrightarrow{\text{ASM}} \hat{\boldsymbol{\theta}} = \hat{\mathbf{W}}^\top \boldsymbol{\theta}} \tag{20}$$

where $\hat{\mathbf{W}}$ is the first $m$ columns of $\mathbf{W}$ and usually there is $m \ll n$.

It has been proved that the error estimation of approximation of $f(\boldsymbol{\theta})$ by $\tilde{f}(\hat{\boldsymbol{\theta}})$ can be formulated by [41]

$$\mathbb{E}\left[\left(f-\tilde{f}\right)^2\right] \leq \mathcal{L}\left(\lambda_{m+1}+\lambda_{m+2}+\cdots\lambda_n\right) \tag{21}$$

where $\mathcal{L}$ is the Lipschitz constant that is only dependent on the domain $\Omega$ and the joint PDF $p_{\boldsymbol{\Theta}}$.

**Remark 1**: The approximation of the gradient $\nabla_{\boldsymbol{\theta}} f$ and the approximation of the integral in Eq. (17) are usually hard to be computed. Especially when the sample size is limited to a small order, the ASM can not be easily employed by some traditional means, e.g., regression-based method [23, 30] and finite difference-based method [42], if there is no any access for additional model evaluations. To this end, a gradient-free method via the Grassmann Gauss-Newton algorithm synthesized with the PCE [36] and the GF-discrepancy minimization algorithm [35] will be adopted to overcome this issue. Discussions will be continued in Section 3, and more details are summarized in Appendix B and Appendix C.

### *2.4. A short discussion on the NARX, POD and ASM*

As discussed above, both NARX and POD can detect the "feature outputs" of a low dimensionality, which specifically results in the NARX coefficients $\boldsymbol{\vartheta} \in \mathbb{R}^{n_g}$ in Eq. (12) and in the POD vector $\hat{\boldsymbol{x}} \in \mathbb{R}^{n_t}$ in Eq. (16), as illustrated in Fig. 1. In general, for both NARX and POD the dynamical property and the data quality are of certain importance. Nonetheless, in our study it seems that the dynamical property may be the dominant key to the NARX, while for the POD the data quality is the key. Besides, a valuable result in the following studied examples will further show that, the data with good quality but relatively small quantity can possibly behave better than the data with poor quality but relatively large quantity.

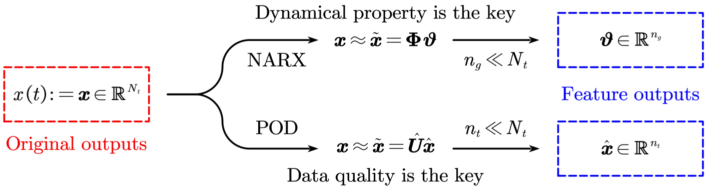

Dynamical property is the key
$\boldsymbol{x} \approx \tilde{\boldsymbol{x}} = \boldsymbol{\Phi}\boldsymbol{\vartheta}$
NARX
$n_g \ll N_t$
$\boldsymbol{\vartheta} \in \mathbb{R}^{n_g}$
Feature outputs
$x(t) := \boldsymbol{x} \in \mathbb{R}^{N_t}$
Original outputs
POD
$n_t \ll N_t$
$\boldsymbol{x} \approx \tilde{\boldsymbol{x}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{x}}$
$\hat{\boldsymbol{x}} \in \mathbb{R}^{n_t}$
Data quality is the key

Figure 1: Detecting the feature outputs via NARX and POD.

The so-called "feature" means that the original outputs (QoIs), which are of high dimensionality, can be well represented in a low-dimensional space under a given small tolerance —for NARX the mean relative error in Eq. (11) and for POD the small value in Eq. (14). Both NARX and POD work well for this purpose, but they have distinct differences on the following concepts:

(1) **Physical significance**. The physical significance of NARX has been widely noticed [31, 33]. For instance, in [33] the authors chose a polynomial type of NARX terms with a maximal order 3, which is

according to their prior knowledge on the cubic nonlinear behaviours of the studied system:

$$\begin{cases} m_s\ddot{x}_1(t) = -k_s\left(x_1(t) - x_2(t)\right)^3 - c\left(\dot{x}_1(t) - \dot{x}_2(t)\right) \\ m_u\ddot{x}_2(t) = k_s\left(x_1(t) - x_2(t)\right)^3 + c\left(\dot{x}_1(t) - \dot{x}_2(t)\right) + k_u\left(z(t) - x_2\right) \end{cases} \tag{22}$$

where $x_1(t)$ and $x_2(t)$ are the QoIs while other parameters are random or deterministic as explained in [33]. The physical interpretation of POD is even more straightforward. As is discussed in [43], the proper orthogonal modes by POD represent the principal axes of inertia of the data, and particularly for linear systems these proper orthogonal modes are converged to the linear normal modes.

(2) **Applied complexity**. Undoubtedly, the application of NARX or POD is much more efficient than the direct computation of the original complex models. Nevertheless, on some specific conditions NARX is more efficient than POD, while in terms of the model construction POD is much simpler than NARX, and vice versa. For example:

1. If one has plenty of knowledge on the nonlinear behaviours of system, viz., knowing its linear or nonlinear dynamical properties, then setting the candidate NARX terms can be very fast and even perfectly determined at the initial trial, thus the metamodelling via NARX can be efficient and accurate as well. For instance, the absolute velocity is taken as one of the NARX model terms in [33] for the Bouc-Wen case, because it is priorly known that the nonlinear behaviour of the response is related to the absolute velocity. On this condition, the NARX-based metamodel can be well-trained with samples of a small size, since the mostly important information of the model has been already “learned” from one’s prior knowledge on the system.

2. Nonetheless, for some cases there may not exist much informative details about the studied model. In general, it is extremely hard to set proper candidate NARX terms for the NARX modelling. To this end, POD provides a totally data-driven approach that there is no requirement for model selections defined by the user. This becomes a most remarkable advantage of POD but a slight shortage of POD as well, since to obtain a well performed POD model, one needs sufficient samples and thus the size of samples may be prohibitively large.

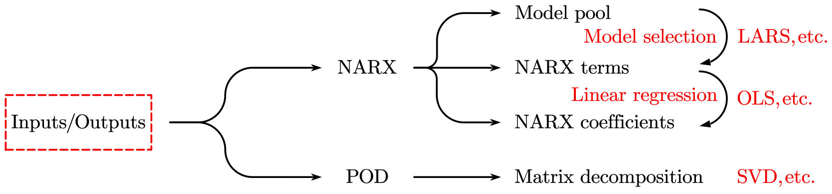


Figure 2: General modelling procedure of NARX and POD.

(3) **Computational efficiency**. The computational efficiency of NARX and POD includes two aspects: the first aspect is on the efficiency of generating the metamodel, and the second one is about the computational efficiency of prediction of responses via the metamodel. The former is summarized in Fig. 2, which shows that the metamodelling of NARX is more complicated than POD. The latter is related to the following two equations, i.e., the prediction modes of NARX and POD:

$$\begin{cases} \text{NARX}: \tilde{x}(t) = \sum_{i=1}^{n_g} \vartheta_i g_i(y(t)) \text{ for } t = 1, 2, \cdots, N_t \Longrightarrow \text{ recursive format} \\ \text{POD}: \tilde{x}(t) = \sum_{i=1}^{n_t} \hat{x}(t)\hat{U}_i(t) \text{ for } t = 1, 2, \cdots, N_t \Longleftrightarrow \text{ matrix format } \tilde{x}(t) = \hat{\boldsymbol{U}}\hat{\boldsymbol{x}} \end{cases} \tag{23}$$

where both NARX and POD models are represented by linear combinations of linear/nonlinear functions with corresponding coefficients. The only difference between NARX and POD is that, NARX terms could be nonlinear while POD modes are all linear[2].

Similar to the NARX and POD, the ASM is aimed at detecting the “feature inputs” of a low dimensionality. The conceptual idea of ASM can be intuitively understood from the perspective of sensitivity analysis, in which it is known that for most issues of UQ on stochastic parametric systems, only a few random variables have the most dominant impacts on the results of interest [29]. In the words of principal component analysis (PCA), there may exist some principal coordinates that reserve the greatest variance of stochastic systems. In ASM, this principal direction is computed by the orthogonal linear transformation as shown in Eq. (20).

## 3. Feature mapping strategy of metamodelling

### 3.1. Basic idea

Inspired by the works in [32, 33] that NARX and POD can perform well on the representation of dynamical systems via relative low-dimensional coefficients, and the works in [36, 41] that ASM can sharply reduce the dimension of input random variables, the following idea comes into being:

*Instead of building a metamodel between the high-dimensional inputs and the high-dimensional outputs, the above-mentioned dimension reduction methods can be synthesized together to provide a compatible feature mapping strategy of metamodelling, i.e., the procedure of metamodelling can be completed between the feature inputs and the feature outputs, of which the dimension may be much lower than the original dimension.*

[2] Herein the “linear” does not mean that the POD mode is a linear function, but oppositely this function is always of nonlinearity. The meaning of “linear” is due to the fact that this function is formatted as numerical values and thus no calculation is of need, which is then considered as a constant vector of high dimensions.

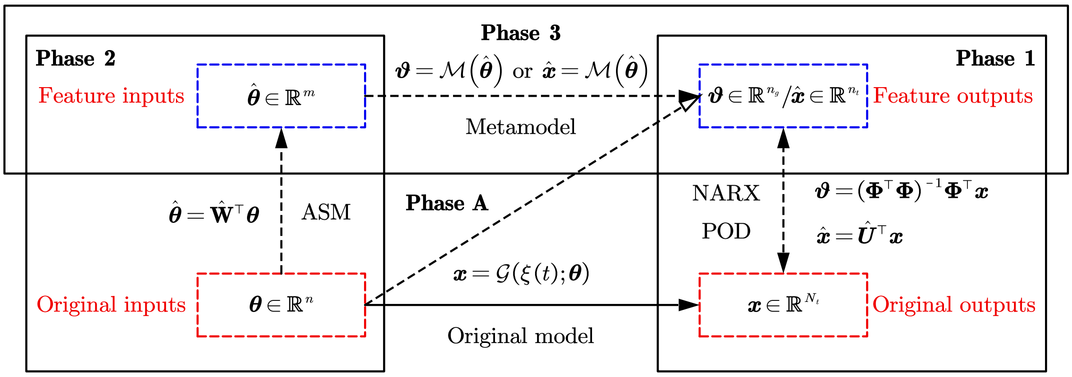

Phase 3
$\boldsymbol{\vartheta} = \mathcal{M}(\hat{\boldsymbol{\theta}})$ or $\hat{\boldsymbol{x}} = \mathcal{M}(\hat{\boldsymbol{\theta}})$
Metamodel
Phase 2
Feature inputs
$\hat{\boldsymbol{\theta}} \in \mathbb{R}^m$
Phase 1
$\boldsymbol{\vartheta} \in \mathbb{R}^{n_g} / \hat{\boldsymbol{x}} \in \mathbb{R}^{n_t}$
Feature outputs
$\hat{\boldsymbol{\theta}} = \hat{\mathbf{W}}^\top \boldsymbol{\theta}$
ASM
Phase A
NARX
POD
$\boldsymbol{\vartheta} = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{x}$
$\hat{\boldsymbol{x}} = \hat{\boldsymbol{U}}^\top \boldsymbol{x}$
$\boldsymbol{x} = \mathcal{G}(\xi(t); \boldsymbol{\theta})$
Original inputs
$\boldsymbol{\theta} \in \mathbb{R}^n$
$\boldsymbol{x} \in \mathbb{R}^{N_t}$
Original outputs
Original model

Figure 3: A feature mapping strategy of metamodelling. Phase 1: Extract feature outputs via NARX or POD; Phase 2: Detect feature inputs via ASM; Phase 3: Metamodel between feature inputs and outputs; Phase A: Auxiliary metamodelling.

The above idea is summarized in Fig. 3, where the proposed feature mapping strategy of metamodelling has three main phases: in **Phase 1** the feature outputs are constructed by NARX or POD; in **Phase 2** the feature inputs are detected via ASM and in **Phase 3** the metamodel between the feature input space and feature output space is built by PC-Kriging method (see Appendix A), and this phase is numerically embedded in Phase 2 (Algorithm 1). It should be emphasized that, any other advanced metamodelling methods in Phase 3 and efficient dimension-reduction methods in Phase 1 and Phase 2 are certainly welcomed and can be possibly adopted in the proposed feature mapping strategy of metamodelling, but these methods must be matched felicitously with each other, especially in numerical calculations. In our current study, the introduced methods ASM, NARX and POD are found to be appropriate for this purpose.

In the proposed feature mapping strategy of metamodelling, the first step is to generate the training point set with a low discrepancy via the GF-discrepancy minimization strategy, which will be discussed later. Then, the model responses are computed correspondingly. With inputs $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_N\}$ and outputs $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ in hand, the purpose of Phase 1 is to extract the feature outputs $\hat{\boldsymbol{x}}$ with a relatively low dimensionality. In this paper, the NARX and POD are suggested to to this work but certainly other methods, if applicable, can also be employed. A self-validated procedure using existing samples is proposed to verify whether the original outputs can be well reconstructed in the range of tolerance error.

For each feature output, the feature inputs are detected via ASM in Phase 2. This procedure is enhanced by the aforementioned GF-discrepancy minimization algorithm and the Grassmann Gauss-Newton algorithm as well. Moreover, an auxiliary **Phase A** is available in Fig. 3 (the diagonal procedure) to adaptively judge whether the feature mapping strategy works better or worse (sometimes possibly), which makes the proposed strategy of metamodelling be more flexible and robust.

### *3.2. Accelerating techniques in Phase 2*

As discussed above, calculating Eq. (17) in Phase 2 is always troublesome. Therefore, the GF-discrepancy minimization algorithm and the Grassmann Gauss-Newton algorithm will be synthesized to solve this issue. The GF-discrepancy minimization algorithm can generate a low-discrepancy point set even when the sample size is relatively low [35], and most importantly, it has been mathematically proved that [44], the discrepancy of the selected point set is independent on the dimension $n$ of $\boldsymbol{\Theta}$, i.e.,

$$\mathcal{D}_{\mathrm{GF}}(\mathcal{P}_n) \leq \mathcal{D}_{\mathrm{EF}}(\mathcal{P}_n) \leq 3.635\mathcal{D}_{\mathrm{GF}}(\mathcal{P}_n) \tag{24}$$

which makes it naturally suitable for high-dimensional issues. While the Grassmann Gauss-Newton algorithm suggested in [36] is a gradient-free method to construct the gradient term in Eq. (17), and there are no additional model evaluations are required. To avoid lengthiness here, the introduced two techniques are provided in Appendix  B and Appendix  C in detail.

The proposed feature mapping strategy is summarized in form of pseudo codes as follows.

**Algorithm 1** Feature mapping strategy of metamodelling

```
Require:
    Select representative points, ${\boldsymbol{\theta}_k \in \mathbb{R}^n}_{k=1}^N \sim p_{\boldsymbol{\Theta}}$    ▷ GF-discrepancy algorithm in Appendix B
    Set a tolerance relative error, $\epsilon = 0.01$
Ensure:
    Metamodel of the feature input-output space, $\hat{\boldsymbol{x}} = \mathcal{M}(\hat{\boldsymbol{\theta}}, t)$ and/or $\boldsymbol{\vartheta} = \mathcal{M}(\hat{\boldsymbol{\theta}}, t)$

    Generate representative samples
for $k = 1, 2, \cdots, N$ do
    $\boldsymbol{x}_k \leftarrow \mathcal{G}(\xi_k(t), \boldsymbol{\theta}_k)$    ▷ Eq. (3)
end for
    Phase 1: Extract the feature output space of $\boldsymbol{x}$
$\boldsymbol{\vartheta} \in \mathbb{R}^{n_g} \leftarrow \boldsymbol{x} \in \mathbb{R}^{N_t}$ where $n_g \ll N_t$    ▷ NARX in Eq. (12)
$\hat{\boldsymbol{x}} \in \mathbb{R}^{n_t} \leftarrow \boldsymbol{x} \in \mathbb{R}^{N_t}$ where $n_t \ll N_t$    ▷ POD in Eq. (16)
    Phase 2: Detect the feature input space of $\boldsymbol{\theta}$    ▷ Take AS-POD for example
for $i = 1, 2, \cdots, n_t$ do
    $\hat{\boldsymbol{\theta}}^{(i)} \in \mathbb{R}^{m_i} \leftarrow \boldsymbol{\theta} \in \mathbb{R}^n$ where $m_i \le n$    ▷ ASM in Eq. (20) and Appendix C
    Phase 3: Build the metamodel and its relative error for prediction    ▷ PC-Kriging in Appendix A
    $\hat{x}_i \approx \tilde{x}_i = \mathcal{M}_i^{\text{AS-POD}}(\hat{\boldsymbol{\theta}}^{(i)})$ and $\epsilon^{\text{AS-POD}} \leftarrow \|\hat{x}_i - \tilde{x}_i\|_2 \,/\, \|\hat{x}_i\|_2$
    $\hat{x}_i \approx \tilde{x}_i = \mathcal{M}_i^{\text{POD}}(\hat{\boldsymbol{\theta}}^{(i)})$ and $\epsilon^{\text{POD}} \leftarrow \|\hat{x}_i - \tilde{x}_i\|_2 \,/\, \|\hat{x}_i\|_2$    ▷ Phase A
    if $\epsilon^{\text{AS-POD}} < \epsilon^{\text{POD}}$ then
        $\mathcal{M}_i \leftarrow \mathcal{M}_i^{\text{AS-POD}}$
    else
        $\mathcal{M}_i \leftarrow \mathcal{M}_i^{\text{POD}}$
    end if
    if $\min\{\epsilon^{\text{AS-POD}}, \epsilon^{\text{POD}}\} > \epsilon$ then
        Unconverged, break
    end if
end for
```

## 4. Numerical applications

### *4.1. Example 1: A 2-dimensional mathematical function*

A 2-dimensional mathematical function is designed herein to illustrate the proposed method:

$$X(t) = f(\Theta_1, \Theta_2) \cdot \xi(t), \ \ t \in [0, 1] \tag{25}$$

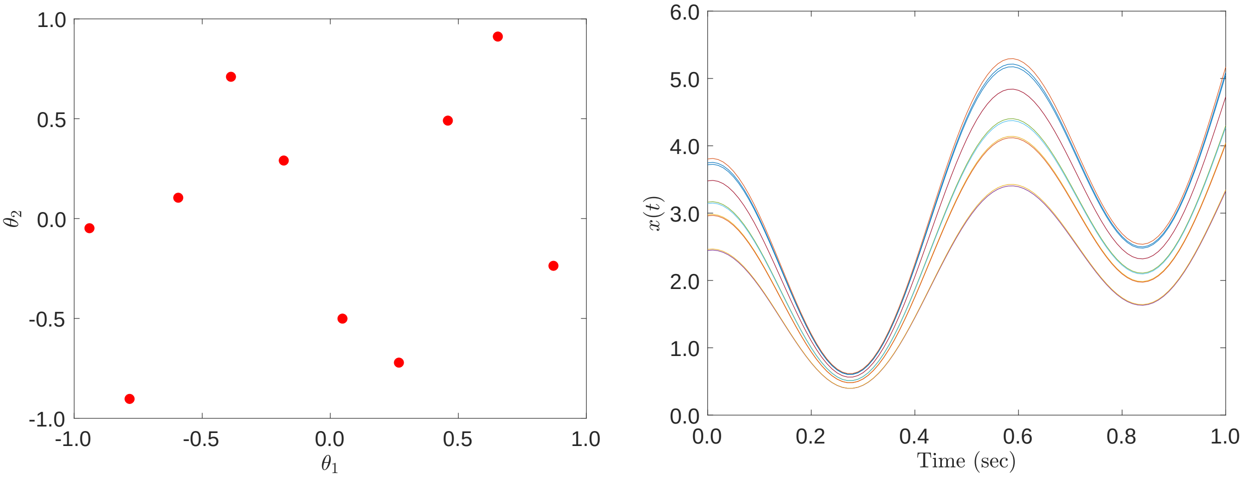
where

$$\begin{cases} \xi(t) = \exp(At) + \cos(\omega t), \\ f(\Theta_1, \Theta_2) = 2 + 0.5(\Theta_1 - \Theta_2)^4 - (\Theta_1 - \Theta_2)^2 - 0.1 \exp(\Theta_1 - \Theta_2) \end{cases} \tag{26}$$

where $A = 1$, $\omega = 3.5\pi$, and $\Theta_i$ for $i = 1, 2$ is independently and uniformly distributed in the interval $[-1, 1]$.

Let the time step size $\Delta t = 0.01$, then we have $x \in \mathbb{R}^{N_t}$ where $N_t = 101$, and the discrete time series are $[0, 0.01, 0.02, \cdots, 1]$. As stated above, instead of building a metamodel between $[\theta_1, \theta_2] \in \mathbb{R}^2$ and $x \in \mathbb{R}^{101}$, the NARX or POD is firstly applied to extract the feature output space (Phase 1), and then detect the feature input space (Phase 2).

Another challenge here is that only 10 samples[3] are assumed to be available for the construction of metamodel, as drawn in Fig. 4. The following will show that under such an extreme condition, i.e., with only sparse data, the proposed feature mapping strategy can still service well for metamodelling while the other methods may not.



(a) The optimal selected 10 samples via [35]

(b) The corresponding sample processes

Figure 4: The size of samples in Example 1 is limited to 10.

[3] The point set is optimally selected with a minimal GF-discrepancy, of which details can be seen in [35] and Appendix B.

### *4.1.1. Extracting the feature output space via NARX*

In this case, the $\xi(t)$ is known as the exogenous input, and from Eq. (25) one is easy to define a set of candidate NARX terms in form of polynomial types, e.g., $g_i(t) \in \left\{\xi^{l_\xi}(t-n_\xi)x^{l_x}(t-n_x)\right\}$ for $l_\xi \in \{0,1,2,3\}$, $l_x \in \{0,1,2,3\}$, $n_\xi \in \{0,1,2,3,4\}$ and $n_x \in \{1,2,3,4\}$. Then, the LARS suggested in [33] is adopted here to pick up the most relevant NARX terms, under a given tolerance mean error, e.g., $\bar{\epsilon} = 10^{-6}$.

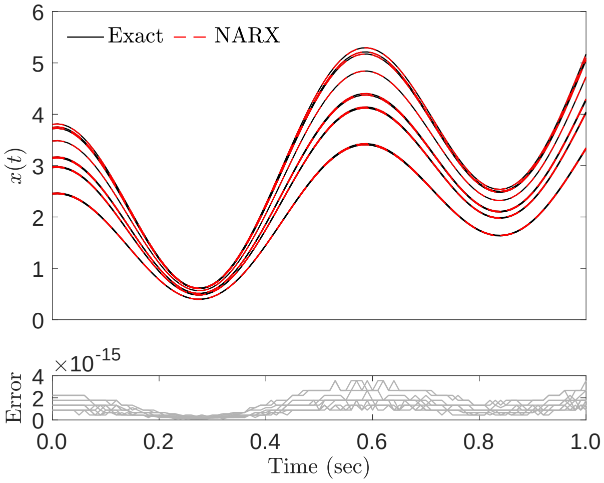


Figure 5: NARX model test in Example 1.

Only one NARX term, i.e., the constant term is extracted to satisfy the condition, which is obvious from Eq. (25) where $x(t)$ is linear to $\xi(t)$. Besides, the recursive formula of NARX for prediction also works well according the test with these 10 samples, as is shown in Fig. 5. Note that now from $x \in \mathbb{R}^{101}$ we get the feature output $\hat{x} \in \mathbb{R}^1$, which is exactly the coefficient $f(\theta_1, \theta_2)$ in Eq. (25). Denote $\hat{x}^{\text{NARX}}$ as the feature output $\hat{x}$ via the method of NARX, such that

$$x \in \mathbb{R}^{101} \Rightarrow \hat{x}^{\text{NARX}} \in \mathbb{R}^1 \tag{27}$$

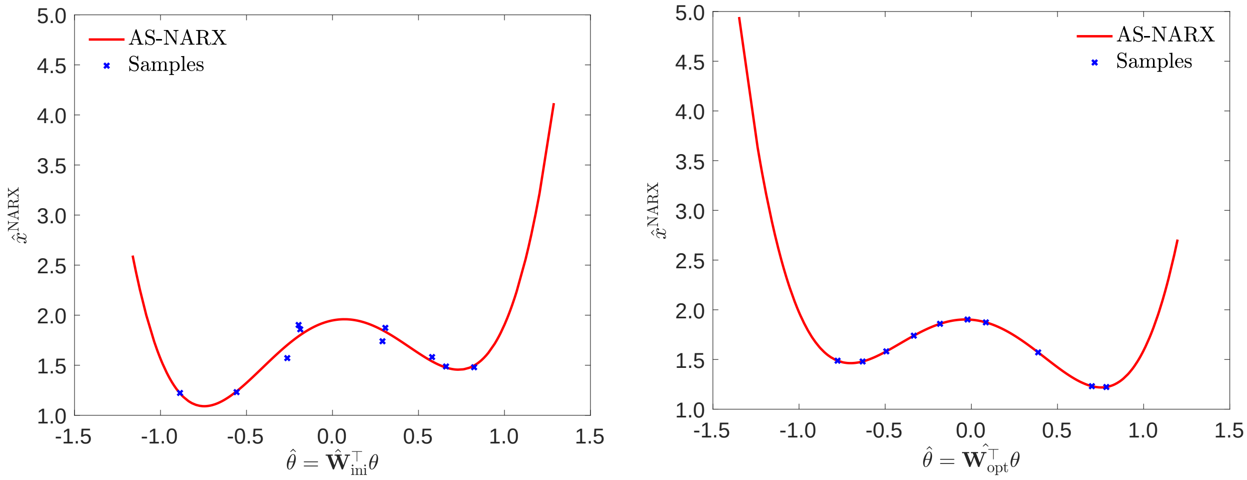
### *4.1.2. Detecting the feature input space (AS-NARX)*

In Phase 2, the feature input space will be detected in order to reduce the dimensionality of the original inputs:

$$\boldsymbol{\theta} \in \mathbb{R}^2 \Rightarrow \hat{\theta} \in \mathbb{R}^1 \tag{28}$$

It should be emphasized that, since the dimension of inputs is 2, there is usually no need to execute Phase 2 to reduce the dimension from 2 to 1 —a bunch of metamodels, e.g., Kriging, PCE and SVM, etc., could provide good results. Nevertheless, remember that there are only 10 samples for metamodelling in this case, so these classical methods may fail without process of dimension-reduction. For instance, if PCE is adopted and its degree is set to 4 that is consistent with Eq. (26), the number of PCE terms become

$\binom{4+2}{4} = 15 > 10$ with dimension of 2, while it turns to be $\binom{4+1}{4} = 5 < 10$ with dimension of 1. The significance of dimension reduction is remarkable on this condition.



(a) The initial trial feature input space

(b) The detected feature input space

Figure 6: Detecting the feature input space by ASM via the Grassmann Gauss-Newton algorithm where the feature out space is extracted by NARX. The feature mapping strategy here is shortly named by AS-NARX.

The initial trial feature input space, as shown in Fig. 6a, is generated by a random but orthogonal matrix, i.e., $\hat{\mathbf{W}}_0 = [-0.8984, 0.4392]^{\top}$. By adopting the Grassmann Gauss-Newton algorithm, the final optimized feature input space is detected by $\hat{\mathbf{W}}_{\text{opt}} = [0.7071, -0.7071]^{\top}$, as shown in Fig. 6b. It can be clearly seen that, a good consistency is achieved between the feature input space $\hat{\theta}$ and the feature output space $\hat{x}^{\text{NARX}}$ by the proposed AS-NARX.

Moreover, as is mentioned in [36], the Grassmann Gauss-Newton algorithm may possibly lead to a local optimization solution, which is mainly due to the initial matrix that is randomly generated. To conquer this issue, 10 trial initial matrices are set to increase the success rate on detecting the feature input space.

### *4.1.3. Extracting the feature output space via POD*

Unlike the NARX, there is no assumption of the POD on any prior knowledge of the exogenous input $\xi(t)$. Therefore, with samples of $x(t)$ we set $\epsilon = 10^{-6}$ as the truncation of the sum of eigenvalues, i.e.,

$$\sum_{j=1}^{n_t} \lambda_j / \sum_{i=1}^{N_t} \lambda_i < 1 - \epsilon \tag{29}$$

The results indicate that the first mode satisfies this condition, which means

$$x \in \mathbb{R}^{101} \Rightarrow \hat{x}^{\text{POD}} \in \mathbb{R}^{1} \tag{30}$$

where $\hat{x}^{\text{POD}}$ stands for the feature output $\hat{x}$ by the POD method.

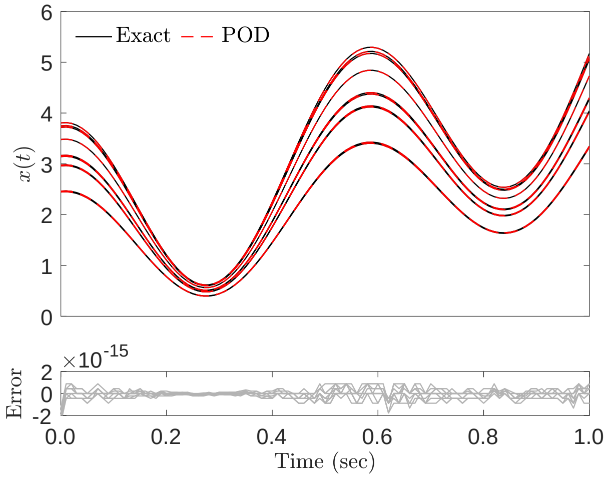
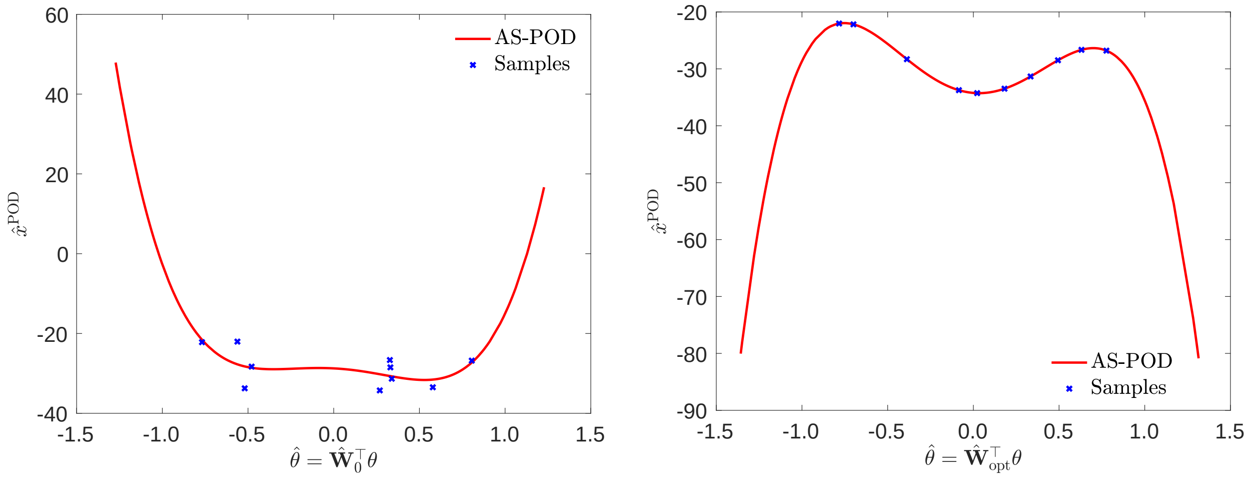


Figure 7: POD model test in Example 1.

Again, the prediction error by POD model appears very small, as shown in Fig. 7.

### 4.1.4. Detecting the feature input space (AS-POD)

In this phase the procedure we need to do is the same as what in Section 4.1.2, and the only difference is that the feature input becomes $\hat{x}^{\text{POD}}$ instead of $\hat{x}^{\text{NARX}}$. The Grassmann Gauss-Newton algorithm is applied again and the results are drawn in Fig. 8. Once again, the results indicate a good convergence of the Grassmann Gauss-Newton algorithm on detecting the optimized feature input space.



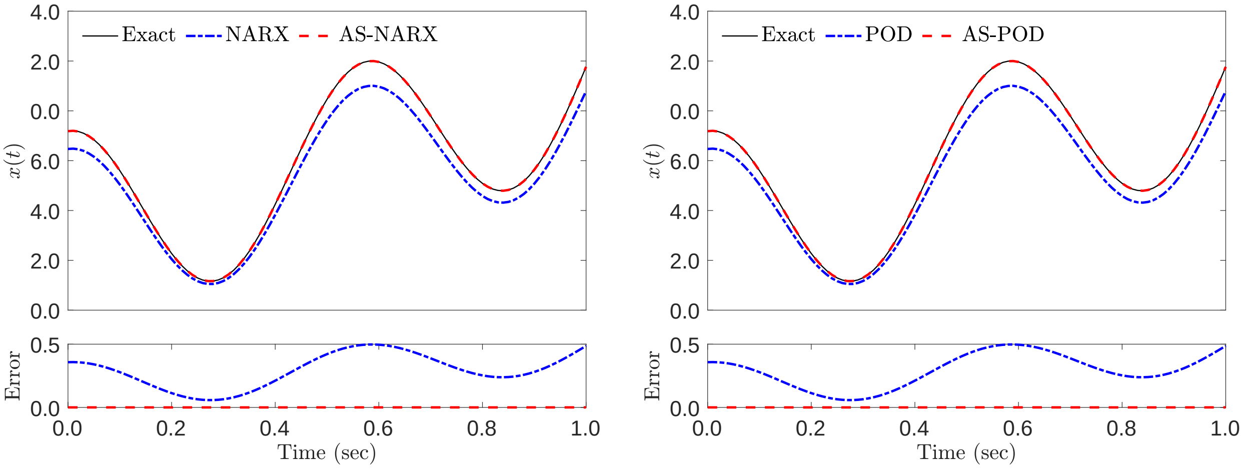
(a) The initial trial feature input space

(b) The detected feature input space

Figure 8: Detecting the feature input space by ASM via the Grassmann Gauss-Newton algorithm where the feature output space is extracted by POD. The feature mapping strategy here in shortly named by AS-POD.

### *4.1.5. Model validation of the proposed AS-NARX and AS-POD*

In this section, the metamodels AS-NARX and AS-POD are validated. To verify the significance of the dimension reduction, the metamodel constructed directly between the original inputs and NARX coefficients (named by "NARX" metamodel), and the one built between the original inputs and POD coefficients (named by "POD" metamodel), are generated for comparison.



(a) Predictions of AS-NARX and NARX metamodels

(b) Predictions of AS-POD and POD metamodels

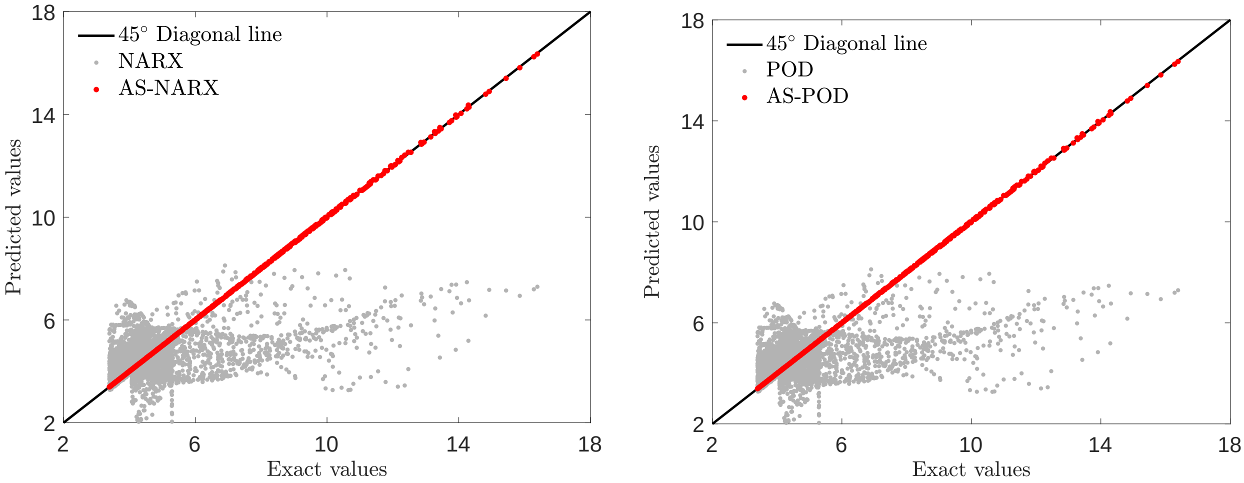
Figure 9: Model validation of the proposed AS-NARX and AS-POD metamodels considering the whole time history.

The Fig. 9 presents the predicted responses by (AS)-NARX and (AS)-POD metamodels compared with actual responses, from which it indicates that the proposed AS-NARX and AS-POD metamodels are of high accuracy for this case.

The accuracy of the proposed AS-NARX and AS-POD metamodels are further validated by MCS of $10^4$ samples. Here the maximal value of responses is taken as the QoI for testing, i.e., $\max\{|x(t)|\}$, as drawn in Fig. 10. Again, the proposed AS-NARX and AS-POD metamodels both exhibit good performance on predicting. Besides, it is interestingly observed that the AS-NARX metamodel behaves nearly the same as the AS-POD metamodel does. The reason for this phenomenon can be explained in Fig. 11: since the feature outputs $\hat{x}^{\text{NARX}}$ is linear to the feature outputs $\hat{x}^{\text{POD}}$, the capacity for prediction is also supposed to be closely similar.

The statistical results of the responses, including the extreme value distribution of $x(t)$, the mean and the standard deviation (std) processes of $x(t)$ are computed. The extreme value distribution of $x(t)$ is presented in Fig. 12, and the mean and the std processes are shown in Fig. 13, correspondingly.

Again, it is observed that the AS-NARX metamodel has the same ability of prediction on the statistical responses of $x(t)$ as the AS-POD metamodel does, which shows a high accuracy compared with the exact



(a) Model validation of AS-NARX and NARX metamodels

(b) Model validation of AS-POD and POD metamodels

Figure 10: Model validation of the proposed AS-NARX and AS-POD metamodels considering the maximal value of responses.

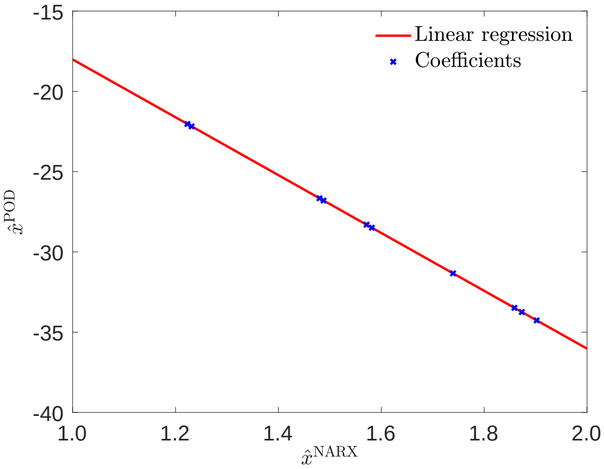
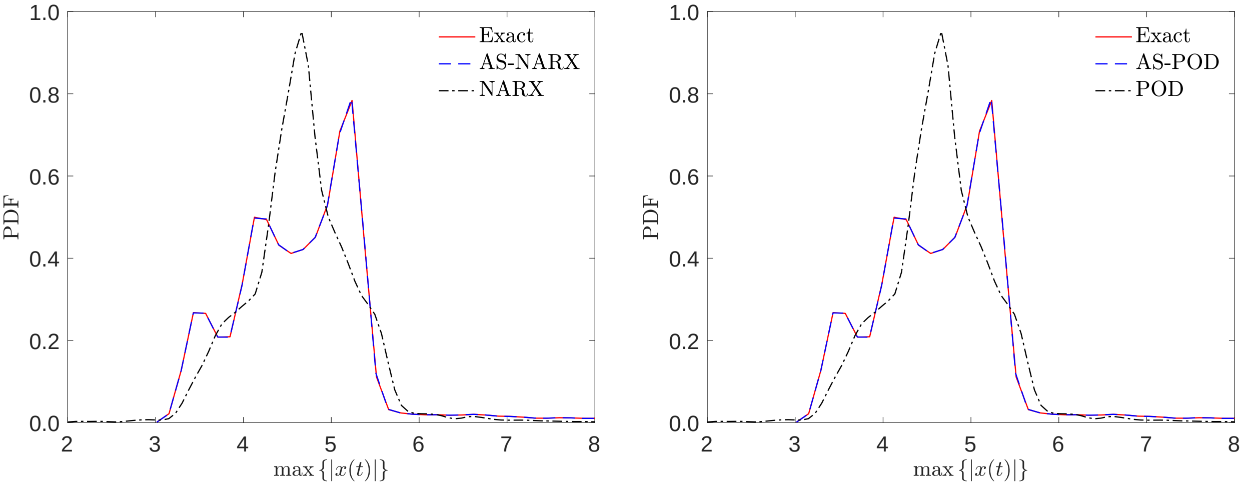


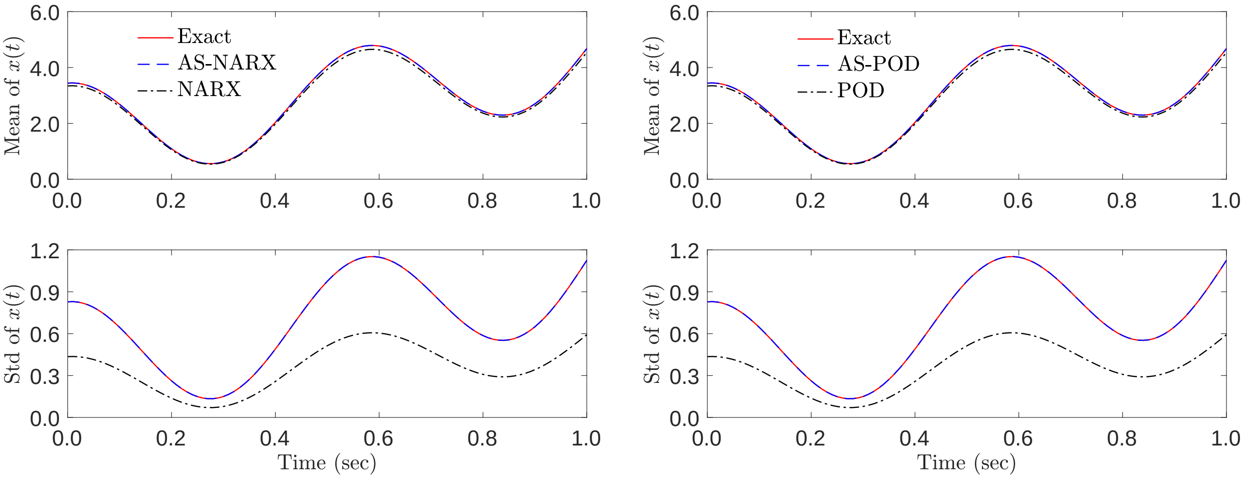
Figure 11: A linear regression between NARX and POD coefficients.



(a) AS-NARX and NARX

(b) AS-POD and POD

Figure 12: Extreme value distribution of $x(t)$ in Example 1.



(a) AS-NARX and NARX

(b) AS-POD and POD

Figure 13: Mean and std processes of $x(t)$ in Example 1.

results. Besides, although the NARX (POD) metamodel presents a very close prediction result on the mean response of $x(t)$, the std response of $x(t)$ by NARX (POD) metamodel is not satisfactory.

### *4.2. Example 2: A quarter car model*

The quarter car model with two degrees of freedom has been studied in [33, 45] as a benchmark problem of UQ on stochastic and nonlinear dynamical models. The system displayed in Fig. 14 is governed by the following ordinary differential equations (ODEs):

$$\begin{cases} m_{\mathrm{s}}\ddot{x}_1(t) = -k_{\mathrm{s}}\left(x_1(t) - x_2(t)\right)^3 - c\left(\dot{x}_1(t) - \dot{x}_2(t)\right) \\ m_{\mathrm{u}}\ddot{x}_2(t) = k_{\mathrm{s}}\left(x_1(t) - x_2(t)\right)^3 + c\left(\dot{x}_1(t) - \dot{x}_2(t)\right) + k_{\mathrm{u}}\left(\xi(t) - x_2(t)\right) \end{cases} \tag{31}$$

where $x_i$, $\dot{x}_i$ and $\ddot{x}_i$ for $i = 1, 2$ are the $i$-th displacement, velocity and acceleration, respectively.

The exogenous input $\xi(t)$ is defined in a trigonometric form as

$$\xi(t) = A\sin\left(\bar{\omega}t\right) \tag{32}$$

with the amplitude $A = 1$ and the mean circular frequency $\bar{\omega} = \frac{1}{10}\sum_{i=1}^{10}\omega_i$, where $\omega_i$ is the $i$-th realization that independently and uniformly distributes in the interval $[0.9\pi, 1.1\pi]$, thus the uncertain inputs are denoted by $\boldsymbol{\Theta} = [\omega_1, \omega_2, \cdots, \omega_{10}]^\top \in \mathbb{R}^{10}$.

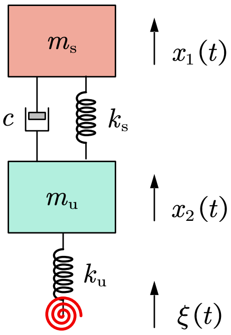


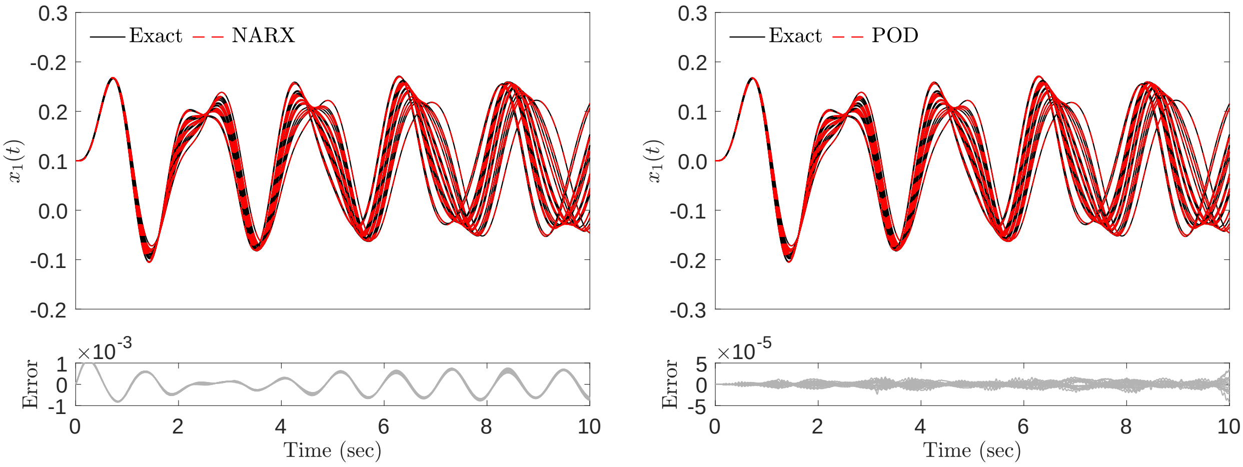
Figure 14: A quarter car model.

The remaining parameters in Eq. (31), i.e., $\{k_{\mathrm{s}}, k_{\mathrm{u}}, m_{\mathrm{s}}, m_{\mathrm{u}}, c\}$ are considered as deterministic input parameters with respect to the mean values $\{2000, 2000, 20, 40, 600\}$ referred to [33]. The time step is taken by $\Delta t = 0.01$ and the time period is $T = 10$ sec, thus the QoI, i.e., the random process $x_1(t)$ is discretized as a random vector of 1001 dimensions. Again, in this example only 20 samples are selected with a minimal GF-discrepancy for metamodelling $\boldsymbol{\theta} \in \mathbb{R}^{10} \mapsto \boldsymbol{x} \in \mathbb{R}^{1001}$.

In this case, the prediction capacity of metamodels by AS-NARX and AS-POD will be quite different. Most importantly, since the uncertainties only originate from the exogenous input $x(t)$ that is assumed to be known in the AS-NARX, therefore, there is a born advantage of AS-NARX compared to AS-POD. We will see this advantage may help the AS-NARX-based metamodel perform better than the AS-POD-based metamodel.

*4.2.1. Phase 1: Feature output space by NARX and POD*

The feature output space that separately extracted by NARX and POD are drawn in Fig. 15.



(a) NARX model test in Example 2

(b) POD model test in Example 2

Figure 15: Testing of the NARX and the POD models in Example 2.

When the NARX is applied, the candidate NARX terms are set as $\{x_1^{l_x}(t-n_x)\xi^{l_\xi}(t-n_\xi)\}$ for $l_x = \{0,1,2,3\}$, $l_\xi = \{0,1\}$, $n_x = \{1,2,3,4\}$ and $n_\xi = \{0,1,2,3,4\}$, where the number of NARX terms is $4 \times 2 \times 4 \times 5 = 160$ (of which the number of unrepeatable terms is 78). To avoid model selections of $2^{78}$ times, the LARS algorithm [38] is adopted and the tolerance mean error is set as $3 \times 10^{-8}$. When the POD is employed, the tolerance error is defined by $5 \times 10^{-4}$ to determine the optimal reduced dimension.

Finally, the feature outputs represented by NARX coefficients are $\hat{x}^{\text{NARX}} \in \mathbb{R}^{10}$, i.e., $\xi(t)$, $\xi(t-1)$, $\xi(t-2)$, $x_1(t-1)$, $x_1(t-4)$, $x_1(t-1)\xi(t)$, $x_1(t-4)\xi(t-4)$, $x_1^2(t-1)\xi(t)$, $x_1^2(t-4)\xi(t-4)$, $x_1^3(t-1)$. While the feature outputs represented by POD coefficients are $\hat{x}^{\text{POD}} \in \mathbb{R}^5$. In short, in Phase 1 we have

$$x_1(t) \in \mathbb{R}^{1001} \Rightarrow \begin{cases} \hat{x}^{\text{NARX}} & \in \mathbb{R}^{10} \\ \hat{x}^{\text{POD}} & \in \mathbb{R}^{5} \end{cases} \tag{33}$$

From Fig. 15 it can be seen that although the number of feature outputs of POD (5) is smaller than that of NARX (10), the testing error of POD model is far less than that of NARX model by the order of magnitude of 2.

### *4.2.2. Phase 2: Feature input space by ASM*

The feature input spaces found by ASM for NARX coefficients are drawn in Figs. 16 and 17, while the feature input spaces detected by ASM for POD coefficients are shown in Fig. 18. What needs illustration is that, in order to increase the success rate of the Grassmann Gauss-Newton algorithm, the initial orthogonal matrix is not randomly picked but approximated by the local linear regression method, which is different from [36] and is discussed in Appendix C.

It is interesting to see that, the feature input can be almost linear, quadratic or cubic to each feature output. This may somehow reflect the dynamical properties of the study system according to the ODEs in Eq. (31), where there are linear terms w.r.t. the velocity, and also cubic terms w.r.t. the displacement.

### *4.2.3. Model validation and discussions*

A total number of $10^4$ MCS samples are calculated as the reference to validate the accuracy of the metamodels constructed by AS-NARX and AS-POD. Again, to verify whether the proposed feature mapping strategy works, the "NARX" and "POD" metamodels are built of which the dimension of original input space is unreduced. The maximum of the absolute value of $x_1(t)$, i.e., $\max\{|x_1(t)|\}$ is defined as the QoI for this model validation.

The validation results are summarized in Fig. 19. It can be seen that, both the AS-NARX and the AS-POD metamodels perform well for predictions, compared with the prediction results by NARX and POD metamodels, respectively. Meanwhile, in Figs. 19a and 19b the AS-NARX metamodel seems to be more accurate than the AS-POD metamodel —the predictions by the former are more concentrated along the 45° diagonal line than that by the latter. As mentioned above, because of the prior knowledge on the exogenous input in the methodology of NARX, there is a natural advantage of NARX-based metamodel compared to the POD-based metamodel that is completely driven by data. From Figs. 19c and 19d it is even much clearer to observe that the accuracy of AS-POD metamodel can be improved with the increase of size of samples.

The statistical results of the responses are summarized in Figs. 20 and 21. It seems that the proposed feature mapping strategy works better in AS-POD than AS-NARX. One reason may be that both the AS-NARX and NARX methods assume that the exogenous input is previously known, but the POD method only requires sufficient data (inputs-outputs). Additionally, On this condition, a lower dimensionality of inputs (e.g., $n = 10$ for this case) seems to be more important to the POD than the NARX-based methods.

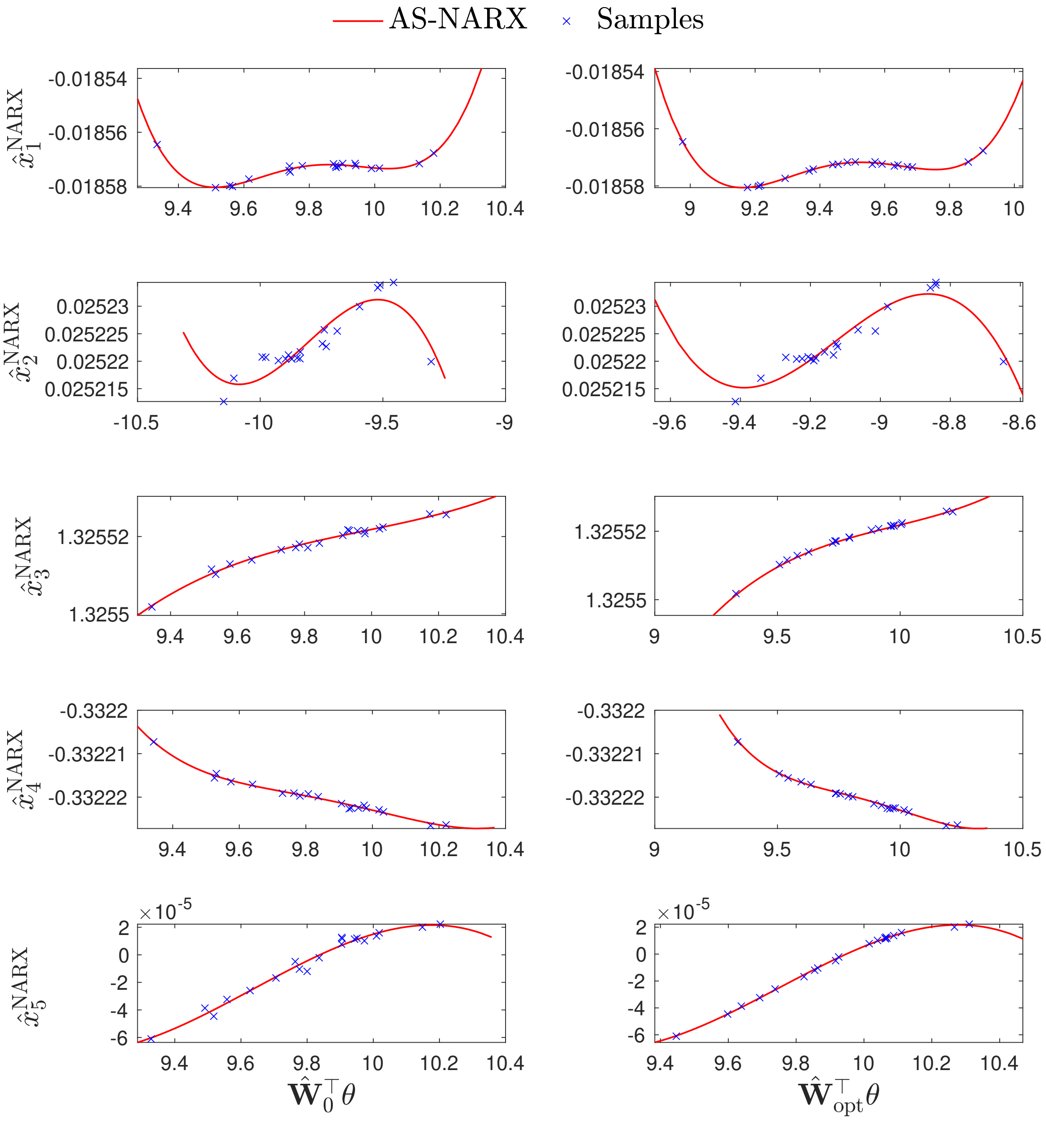


Figure 16: The feature input space of the 1st to the 5th NARX coefficients via the Grassmann Gauss-Newton algorithm. The left columns: the initial feature input space; the right columns: the optimal feature input space.

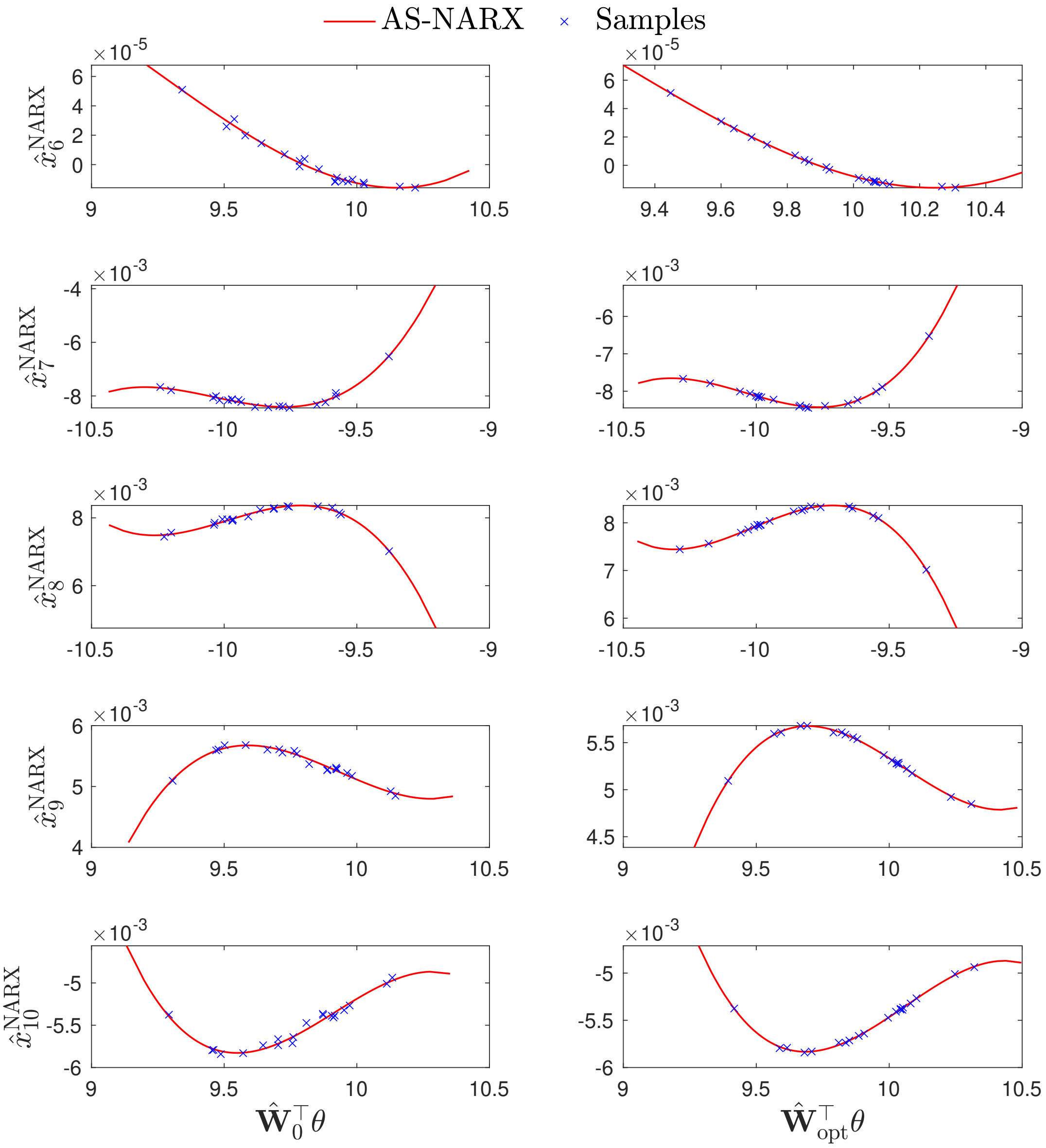


Figure 17: The feature input space of the 6th to the 10th NARX coefficients via the Grassmann Gauss-Newton algorithm. The left columns: the initial feature input space; the right columns: the optimal feature input space.

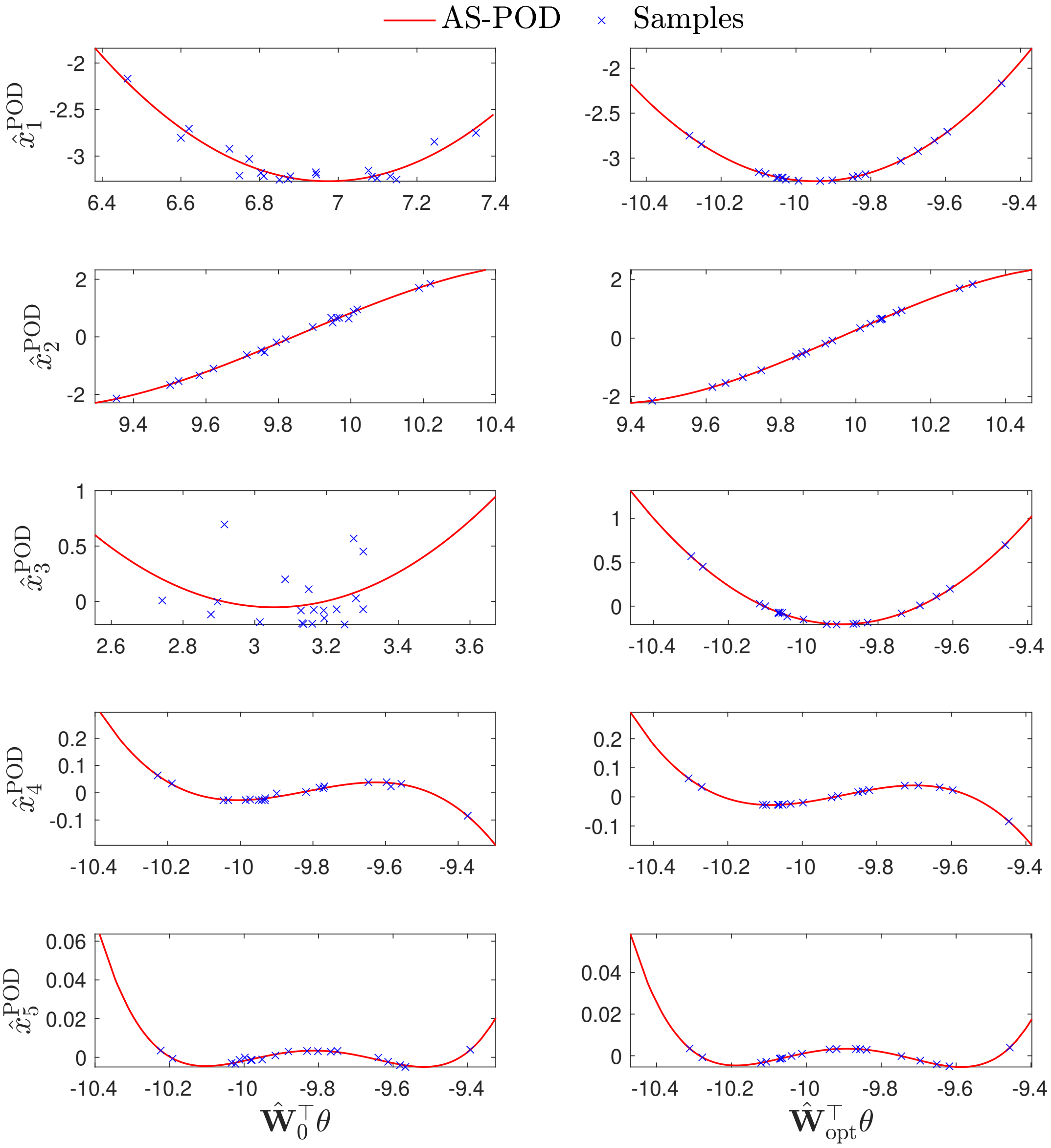
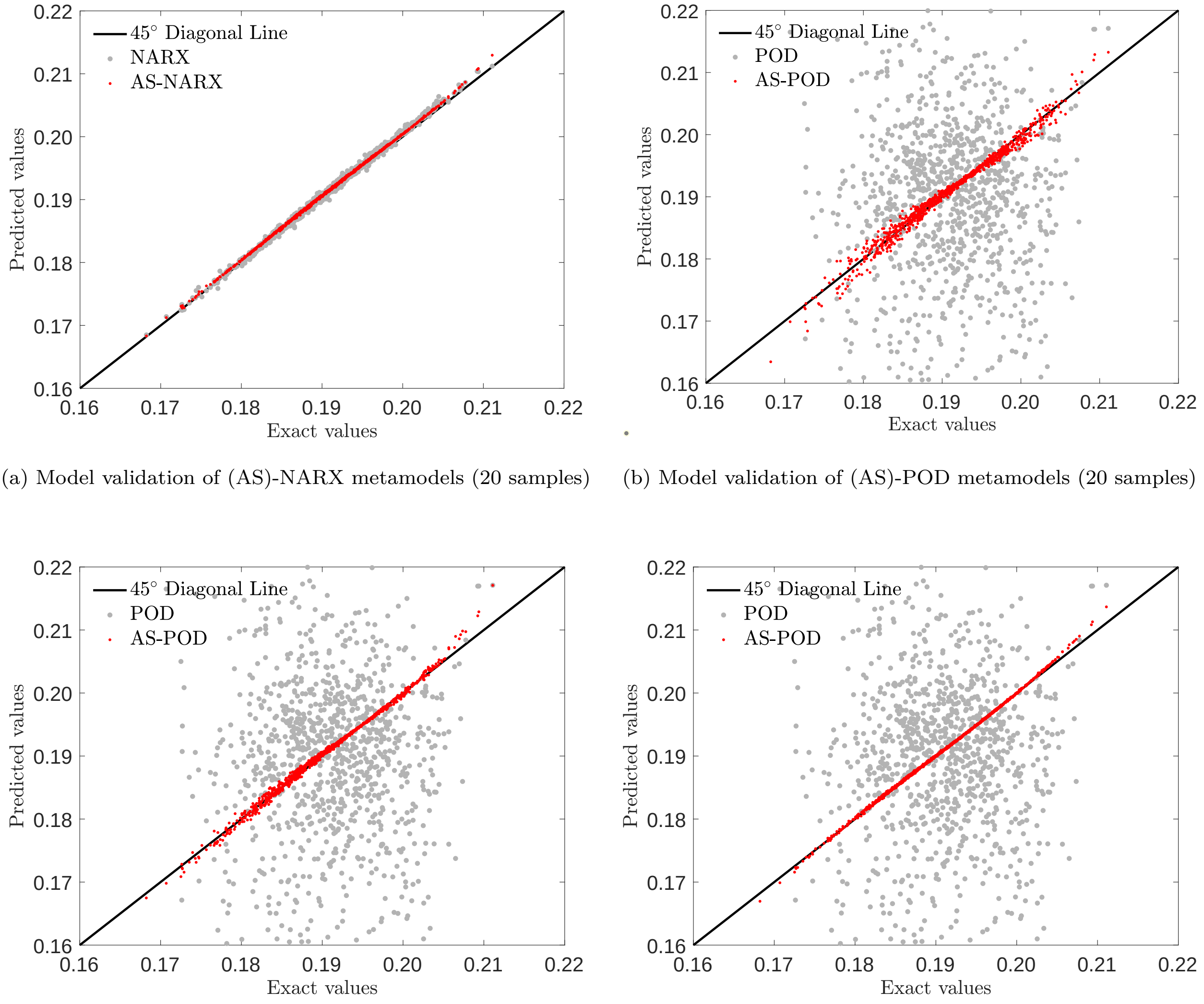


Figure 18: The feature input space of the POD coefficients via the Grassmann Gauss-Newton algorithm. The left columns: the initial feature input space; the right columns: the optimal feature input space.



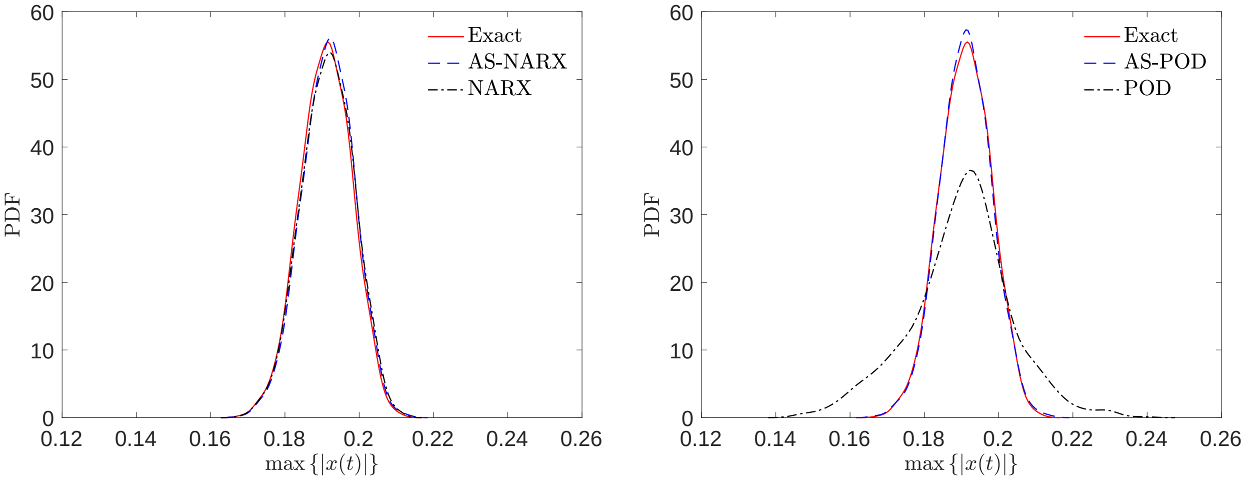
(a) Model validation of (AS)-NARX metamodels (20 samples)

(b) Model validation of (AS)-POD metamodels (20 samples)

(c) Model validation of (AS)-POD metamodels (50 samples)

(d) Model validation of (AS)-POD metamodels (100 samples)

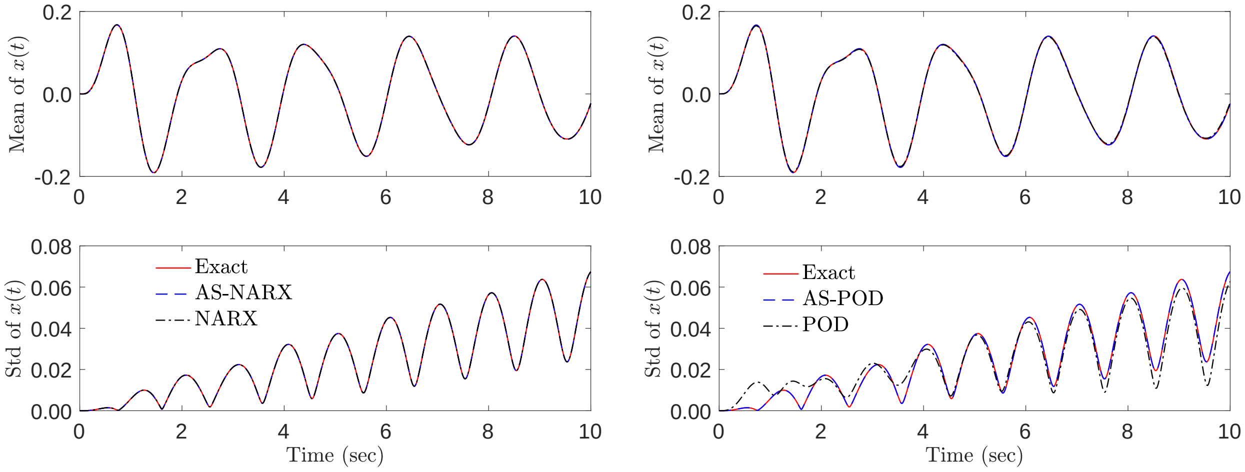
Figure 19: Model validation of the (AS)-NARX and (AS)-POD metamodels in Example 2.



(a) AS-NARX and NARX

(b) AS-POD and POD

Figure 20: Extreme value distribution of $x(t)$ in Example 2.



(a) AS-NARX and NARX

(b) AS-POD and POD

Figure 21: Mean and std processes of $x(t)$ in Example 2.

### *4.3. Example 3: A Bouc-Wen oscillator subject to the El Centro earthquake ground motions*

In this example, we consider the SDOF Bouc-Wen oscillator [33] subjected to the El Centro earthquake ground motions. The governing equation of motion of the system reads:

$$\begin{cases} \ddot{x}(t) + 2\zeta\omega\dot{x}(t) + \omega^2\left(\rho x(t) + (1-\rho)z(t)\right) = -\xi(t) \\ \dot{z}(t) = \gamma\dot{x}(t) - \alpha|\dot{y}(t)||z(t)|^{k-1}z(t) - \beta\dot{x}(t)|z(t)|^k \end{cases} \tag{34}$$

where $\zeta$ and $\omega$ are the damping ratio and the fundamental frequency, respectively. The parameters $\rho$, $\gamma$, $\alpha$, $k$ and $\beta$ govern the hysteric nonlinearity of the system, and $\xi(t)$ is the El Centro earthquake ground motions of North-Western direction. The peak ground acceleration (PGA) is scaled to 0.4g.

The following parameters in the Bouc-Wen model are set to be deterministic: $\zeta = 0.02$, $\rho = 0.8$, $\gamma = 1$, $n = 1$, $\beta = 0$ and $\alpha = 50$. The uncertainties in the considered system is characterized by the $\omega$ that computed by $\omega = \sum_{i=1}^{30}\theta_i/30$, in which $\theta_i$ is the $i$-th realization that independently and uniformly distributes in the interval $[5.373, 6.567]$ referred to [33].

We first select 300 representative samples by the GF-minimization algorithm, and then solve the system of ODEs by means of the MATLAB solver `ode45` with the total duration $T = 20$ sec and the time step $\mathrm{d}t = 0.02$ sec, therefore the responses of the system are denoted by $\boldsymbol{x} \in \mathbb{R}^{1001}$ (the discretized displacement time history) and $\dot{\boldsymbol{x}} \in \mathbb{R}^{1001}$ (the discretized velocity time history $\dot{x}(t)$). The mission now becomes how to build the metamodel between the multiple inputs $\boldsymbol{\theta} \in \mathbb{R}^{30}$ and the stochastic responses $\boldsymbol{x} \in \mathbb{R}^{1001}$.

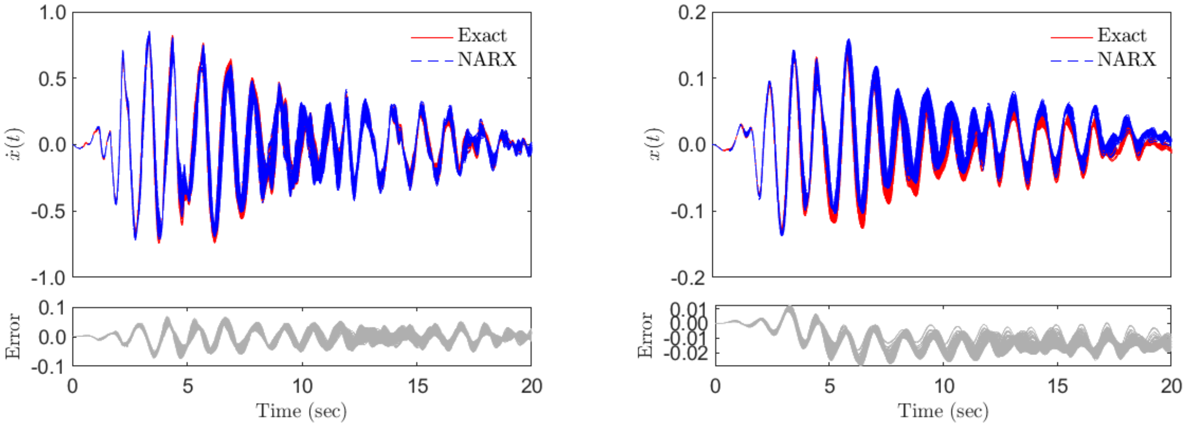
Since the exogenous excitation is previously known, the NARX method seems to be naturally appropriate for extracting the feature output space. The candidate NARX terms are given by $g_k(t) = \xi^l(t-i)|\dot{x}(t-1)|^m$ and $g_k(t) = \dot{x}^l(t-j)|\dot{x}(t-1)|^m$ with $l = \{0,1\}$, $m = \{0,1\}$, $j = \{1,2,3,4\}$ and $i = \{0,1,2,3,4\}$. The use of the velocity time history $\dot{x}(t)$ is due to the hysteric behaviour of the studied Bouc-Wen model in Eq. (34).

The initial number of NARX terms is 19 in total. The most appropriate NARX model that achieves tolerance mean relative error $\bar{\epsilon} = 4 \times 10^{-6}$ over 300 samples contain 12 terms, viz., $\xi(t)$, $\xi(t-1)$, $\xi(t-2)$, $\xi(t-3)$, $\xi(t-4)$, $\xi(t)|\dot{x}(t-1)|$, $\xi(t-1)|\dot{x}(t-1)|$, $\xi(t-2)|\dot{x}(t-1)|$, $\dot{x}(t-1)$, $\dot{x}(t-4)$, $\dot{x}(t-1)|\dot{x}(t-1)|$, $\dot{x}(t-3)|\dot{x}(t-1)|$. This means the feature outputs are now extracted from $\dot{\boldsymbol{x}} \in \mathbb{R}^{1001}$ to $\hat{x}^{\mathrm{NARX}} \in \mathbb{R}^{12}$.

The NARX model testing is complete in Fig. 22, where the displacement time histories are computed by $x(t) = \int_0^t \dot{x}(\tau)\mathrm{d}\tau$. The time histories of error are calculated by the difference between the exact values and the predicated values by NARX model. One observes that the predicted velocity trajectories by NARX model are remaining acceptable, e.g., the maximal relative error no exceeds 0.1, while the displacement ones are relatively well predicted, viz., the maximal relative error no exceeds 0.05.

The mean and std responses of $\dot{x}(t)$ are drawn in Fig. 23a, and Fig. 23b shows the mean error of the AS-NARX metamodel. The mean error over the whole time history is defined by

$$\bar{\epsilon} = \frac{\int_0^T |\dot{x}^{\mathrm{MCS}}(t) - \dot{x}^{\mathrm{AS-NARX}}(t)|\mathrm{d}t}{T} \Big/ \max_{t\in[0,T]} |\dot{x}(t)^{\mathrm{MCS}}| \tag{35}$$

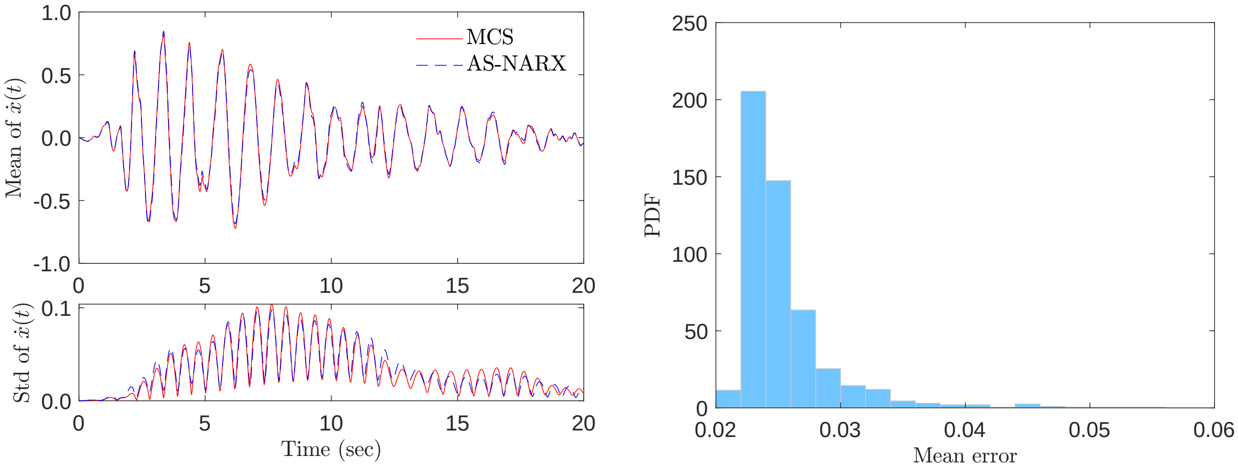


(a) Velocity time histories

(b) Displacement time histories

Figure 22: Testing of the NARX model in Example 3.

where $\dot{x}(t)^{\mathrm{MCS}}$ is the exact response of $\dot{x}(t)$ by MCS and $\dot{x}(t)^{\mathrm{AS-NARX}}$ is the predicted response of $\dot{x}(t)$ by AS-NARX, respectively. The mean errors in Fig. 23a for the mean and std responses are 0.0222 and 0.0437, and the distribution of mean errors over $10^4$ simulations is presented in Fig. 23b where it can seen that the mean error mostly distributes in the interval $[0.02, 0.03]$. The above results indicates a good accuracy of the proposed AS-NARX metamodelling method.



(a) Mean and std responses of $\dot{x}(t)$

(b) Histogram of mean error

Figure 23: The statistical results of Example 3.

### *4.4. Example 4: A subsystem of the NASA UQ challenge 2019*

Since 2019, the NASA Langley Challenge on Optimization under Uncertainty [46] developed by the NASA Langley Research Center, has built a global community of researchers towards some common issues of UQ. This challenge problem is written in a discipline-independent framework, and it is distributed as content-obscured p-code MATLAB files (e.g., yfun.m), viz., as a "black box" to prevent any prior knowledge on complex dynamic systems.

In this example, the first sub-system of the challenge problem (computed by the function yfun.m) is our studied system of interest, which is formulated by

$$x(t) = x(\boldsymbol{a}, \boldsymbol{e}, t) \tag{36}$$

where the symbol $x(t)$ is still used to represent the high-dimensional outputs, $\boldsymbol{a} \in [0, 2]^5$ and $\boldsymbol{e} \in [0, 2]^4$ are structure-unknown aleatory and epistemic variables. Since the present paper only focuses on metamodelling between high-dimensional inputs and outputs (not the all UQ challenge problems), the following assumptions will be made in this example:

(1) The epistemic uncertainty $\boldsymbol{e}$ is simplified to be $[0, 0, 0, 0]$, while the aleatory uncertainty $\boldsymbol{a}$ is redefined by $\boldsymbol{a} = [\bar{a}, \bar{a}, \bar{a}, \bar{a}, \bar{a}]$ with $\bar{a} = \frac{1}{100}\sum_{i=1}^{100}\theta_i$, where $\theta_i$ is the $i$-th realization that independently and uniformly distributes in $[0, 2]$.

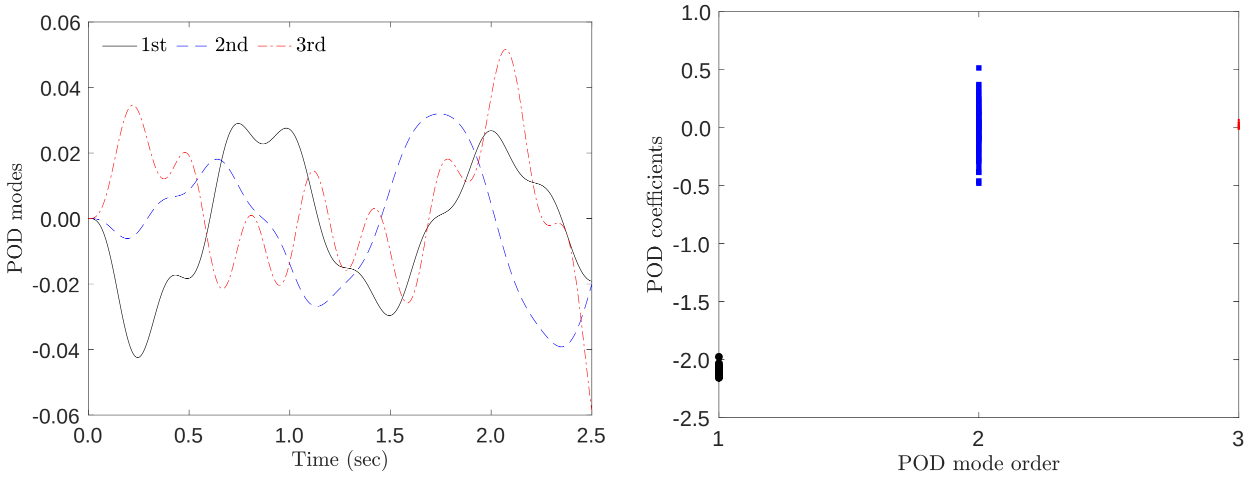
(2) The studied time period is $T = 2.5$ sec with time step $\Delta t = 10^{-3}$.

In summary, the studied system in Eq. (36) now becomes $\boldsymbol{\theta} \in \mathbb{R}^{100} \mapsto \boldsymbol{x} \in \mathbb{R}^{2501}$. Only 200 samples are generated by the GF-discrepancy minimization algorithm, and the POD is applied to extract the feature outputs from the calculated responses, since it is impossible to know the physical properties of the system.

In current study, the dimension of feature outputs is found to be 3 to satisfy a given tolerance error $10^{-7}$ by POD, and the corresponding POD modes and coefficients are drawn in Fig. 24. A comparison between the original responses and the reconstructed responses is shown in Fig. 25, which indicates a relatively high accuracy of dimension-reduction via POD.

The feature inputs are detected by AS-POD as shown in Fig. 26. Once more, the feature inputs w.r.t. the feature outputs are successfully detected by ASM, even the initial feature input is not a good trial, i.e., an approximated active subspace computed by local linear regression method (Appendix C). The statistical results are presented in Fig. 27, which shows a good agreement with the exact solutions.

Testing work is presented in Fig. 28 taking an arbitrary prediction for example, and in Fig. 29a model validation is done by using $10^4$ MCS samples where the absolute maximal value of the response is of interest. Apparently, the proposed AS-POD can perform better than the POD in aspects of global trend and local accuracy. Moreover, the case of no use of GF-discrepancy minimization technique is also present in Fig.



(a) POD modes

(b) POD coefficients

Figure 24: The feature outputs extracted via POD in Example 4.

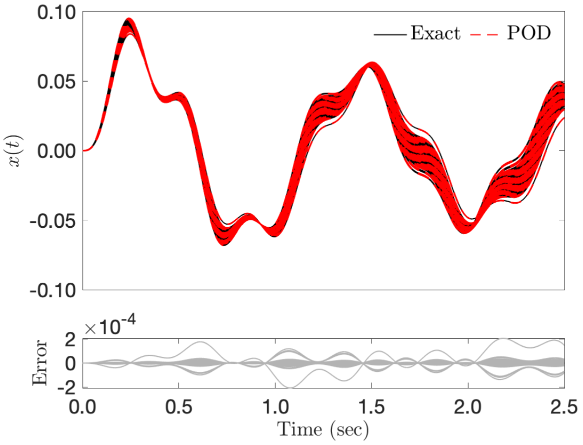


Figure 25: The original responses and the reconstructed responses via POD.

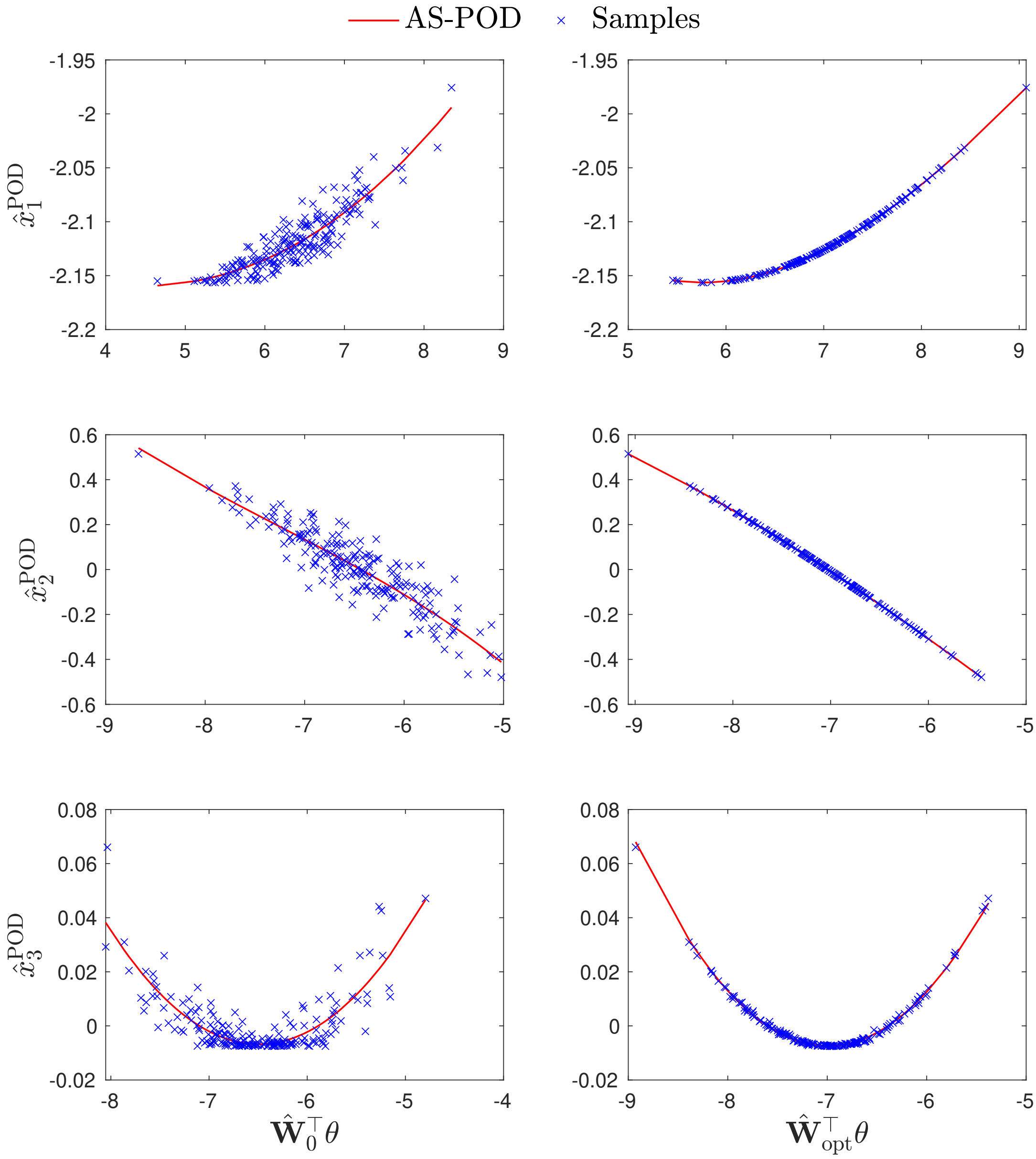
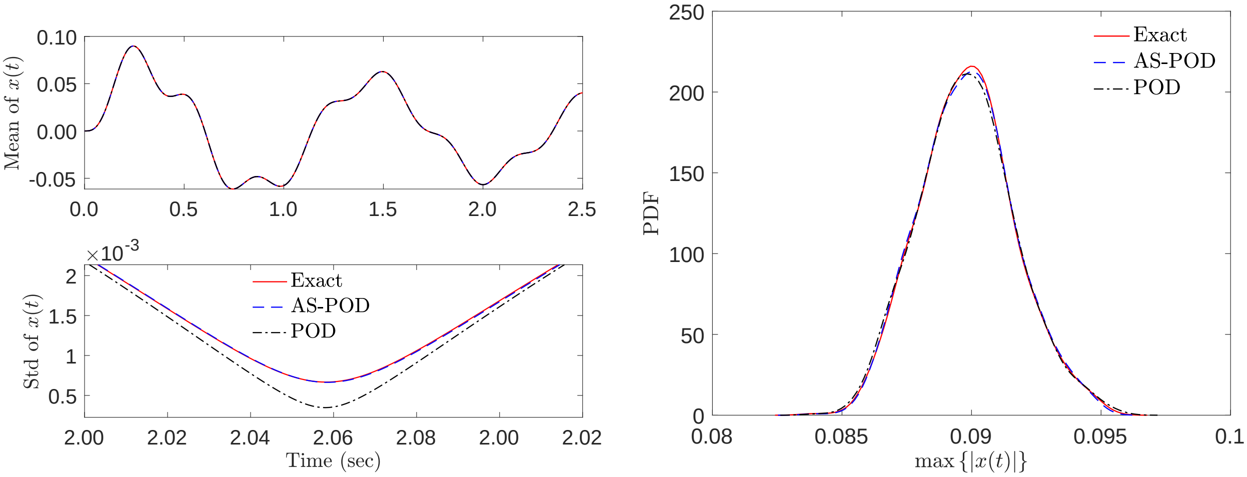


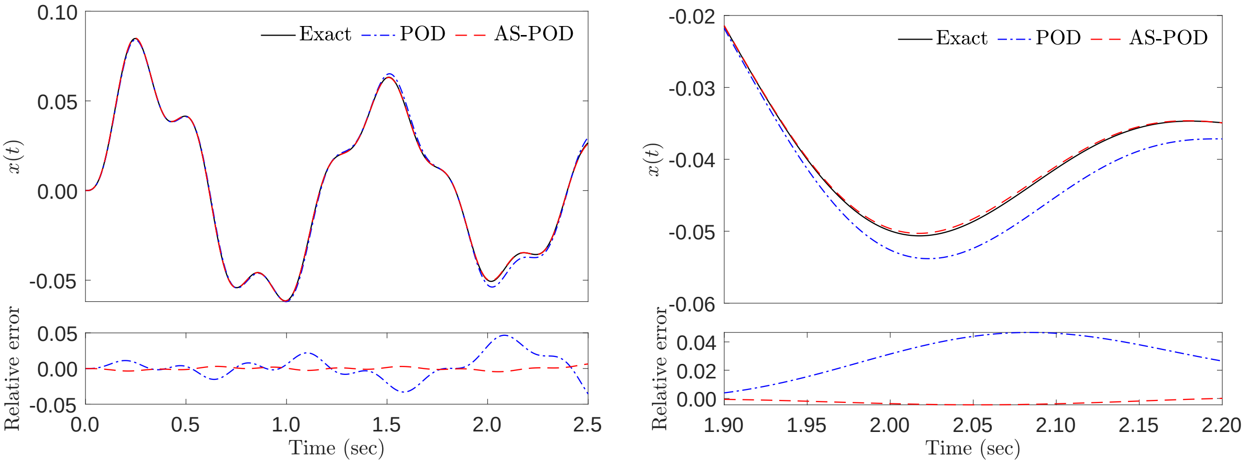
Figure 26: The feature input space of the POD coefficients via the Grassmann Gauss-Newton algorithm. The left columns: the initial feature input space; the right columns: the optimal feature input space.



(a) Mean and std processes of $x(t)$

(b) PDF of max$\{|x(t)|\}$

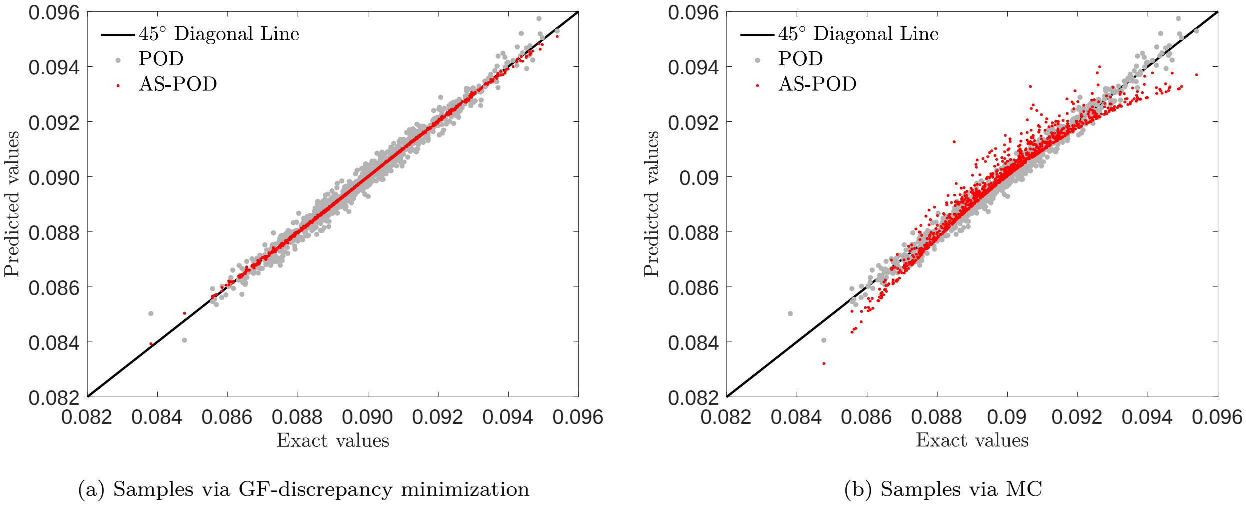
Figure 27: The statistical results of $x(t)$ in Example 4.



(a) Predictions of the whole response

(b) Predictions in a local time interval

Figure 28: Model validation of the proposed AS-POD considering the whole response.



(a) Samples via GF-discrepancy minimization

(b) Samples via MC

Figure 29: Model validation of the (AS)-POD metamodels in Example 4.

29b, which shows a no good prediction when MC samples are utilized. The main reason is that the optimal active subspace is sensitive to the initial trial point when the Grassmann Gauss-Newton algorithm is adopted. When the approximated active subspace is computed as the initial trial point by the local linear regression method, the discrepancy of point set becomes essential since it is related to the integral error in Eq. (17).

Noting that the studied sub-system is based on the NASA UQ challenge issues, in which the dimension of output response is high. In this regard, the proposed mapping feature strategy may also provide some information on uncertainty modelling by combining the Bayesian updating method or others.

## 5. Conclusions

With the steady development in structural engineering, computational model-based analysis and design has taken a central role in modern structural analysis. Nonetheless, computational models, for engineering purposes, may be extremely large and time-consuming. Therefore, metamodels (a.k.a. surrogate models) with performance of fast and high-accurate prediction are gaining much attention, but there is always a challenging issue on metamodelling of stochastic dynamical systems with nonlinear behaviours and high-dimensional uncertainties. To this end, a wide variety of dimension reduction methods are developed to circumvent the issue of “curse of dimensionality”, such as the nonlinear autoregressive with exogenous inputs (NARX), the proper orthogonal decomposition (POD), and the active subspace method (ASM), etc.

In this paper, a novel feature mapping strategy of metamodelling is proposed. The basic idea is to implement a double process of dimension-reduction, i.e., extracting the feature outputs with a much lower dimension than the original outputs of high-dimension, and detecting the feature inputs with a relatively lower dimension than the original inputs of high-dimension. The former part is realized via NARX and POD methods, while the latter procedure is enhanced by ASM. Moreover, to accelerate the convergence of ASM, the Grassmann Gauss-Newton algorithm and the GF-discrepancy minimization strategy are integrated, and the PC-Kriging is adopted to construct the metamodel between the generated feature inputs and feature outputs. The concluding remarks include:

(1) When some of the dynamical properties of the system is previously known, and especially the exogenous excitation is foregone, the NARX may be an ideal method to extract the feature outputs. Whereas, if the dynamical system is considered as a "black box" (e.g., the Example 4), and only the outputs (e.g., dynamical responses) are available, then POD can be an appropriate tool to construct the feature output space in particular with sufficient dataset.

(2) The introduced two techniques can effectively accelerate the convergence of ASM. The GF-discrepancy minimization algorithm decreases the total number of training samples with a relatively low discrepancy, while the Grassmann Gauss-Newton algorithm ensures to find the optimal feature input space without requirement of additional model evaluations.

(3) The studied examples indicate that, the proposed feature mapping strategy can provide an efficient and accurate capacity of metamodelling, especially when the number of model evaluations is limited.

Investigation on the application of the proposed method to more complicated cases is ongoing, e.g., featuring mapping metamodelling of high-rise reinforced concrete structures subjected to stochastic ground motions by introducing the physical random function model of earthquakes, uncertainty quantification with high-dimensional input uncertainties represented by random fields or stochastic processes, and reliability analysis combining the proposed method and the probability density evolution method, etc. Some open issues of the proposed method are remained to be further studied, including, adaptive selection of dimension-reduced methods, optimal decision of dimensions of feature inputs and outputs, and unsupervised learning of metamodelling in feature space, etc.

## Appendix A. PC-Kriging metamodel method

A basic Kriging metamodel can be described by

$$x = \mathcal{G}(\boldsymbol{\theta}) \approx \mathcal{M}^{\mathrm{K}}(\boldsymbol{\theta}) := \boldsymbol{\beta}^{\top}\boldsymbol{g}(\boldsymbol{\theta}) + \sigma^2 G(\boldsymbol{\theta}, \omega) \tag{A.1}$$

where $\boldsymbol{\beta}^{\top}\boldsymbol{g}(\boldsymbol{\theta})$ is the trend function, $\sigma^2$ is the variance and $G(\boldsymbol{\theta}, \omega)$ is a standard stationary Gaussian process with zero mean and unit variance, where $\omega$ represents the underlying probability space that is characterized in terms of a correlation function $R = R(\boldsymbol{\theta}, \boldsymbol{\theta}', \boldsymbol{\alpha})$ and its hyperparameters $\boldsymbol{\alpha}$.

A basic PCE metamodel can be formulated by [12]

$$x = \mathcal{G}(\boldsymbol{\theta}) \approx \mathcal{M}^{\mathrm{PC}}(\boldsymbol{\theta}) := \sum_{\boldsymbol{a} \in \mathcal{A}} c_{\boldsymbol{a}} \Psi_{\boldsymbol{a}}(\boldsymbol{\theta}) \tag{A.2}$$

where $\boldsymbol{a} \in \mathcal{A} \subset \mathbb{N}^M$ are multi-indices, $c_{\boldsymbol{a}}$ are the PC coefficients, and $\Psi_{\boldsymbol{a}}(\boldsymbol{\theta})$ are multivariate orthonormal polynomials.

The PC-Kriging takes advantage of the regression-type PCE to capture the global properties of the model, and also holds the interpolation feature of Kriging to seize the local variations [34]. In other words, PC-Kriging is a specific Kriging metamodel with a PC trend function. Therefore, in this paper we use the combination of PCE and Kriging, i.e., PC-Kriging to improve the efficiency of metamodelling. On the basis of Kriging in Eq. (A.1) and PCE in Eq. (A.2), the expression of PC-Kriging is as follows:

$$x = \mathcal{G}(\boldsymbol{\theta}) \approx \mathcal{M}^{\mathrm{PCK}}(\boldsymbol{\theta}) := \sum_{\boldsymbol{a}\in\mathcal{A}} c_{\boldsymbol{a}} \Psi_{\boldsymbol{a}}(\boldsymbol{\theta}) + \sigma^2 G(\boldsymbol{\theta}, \omega) \tag{A.3}$$

To construct the PC-Kriging metamodel in Eq. (A.3), the optimal set of polynomials $\Psi_{\boldsymbol{a}}(\boldsymbol{\theta})$ and the hyperparameters $\{\sigma^2, \boldsymbol{\alpha}, \boldsymbol{a}\}$ must be determined. In this paper, the MATLAB-based UQLab toolbox [47] (sequential PC-Kriging) is utilized to help build the PC-Kriging metamodel, where the PC part (trend function) is determined by LARS algorithm [38], and then the hyperparameters of PC-Kriging metamodel is computed and calibrated.

## Appendix B. GF-discrepancy minimization algorithm

The GF-discrepancy minimization algorithm [35, 44] adopted in this paper is to generate a set of representative samples with a relatively low discrepancy. The algorithm includes three main steps:

**Step 1.** From a Sobol sequence $\mathcal{S}_N = \{\boldsymbol{u}_q = (u_{1,q}, u_{2,q}, \cdots, u_{n,q})^\top\}_{q=1}^N$ over a unit hypercube, generate an initial point set $\mathcal{P}_N^{(0)} = \{\boldsymbol{\theta}_q^{(0)} = (\theta_{1,q}^{(0)}, \theta_{2,q}^{(0)}, \cdots, \theta_{n,q}^{(0)})^\top\}_{q=1}^N$ by

$$\theta_{i,q}^{(0)} = F_{\Theta_i}^{-1}(u_{i,q}),\ i = 1, 2, \cdots, n \text{ and } q = 1, 2, \cdots, N \tag{B.1}$$

where $F_{\Theta_i}^{-1}(\cdot)$ is the inverse marginal CDF of $\Theta_i$.

**Step 2.** For $i = 1, 2, \cdots, n$, rearrange the point set in each dimension to make points be close to each other:

$$\theta_{i,q}^{(1)} = F_{\Theta_i}^{-1}\left(\sum_{k=1}^{N} \frac{1}{N} \cdot I\left(\theta_{i,k}^{(0)} < \theta_{i,q}^{(0)}\right) + \frac{1}{2} \cdot \frac{1}{N}\right),\ q = 1, 2, \cdots, N \tag{B.2}$$

where $I(A)$ is the indicator function equal to 1 if and only if $A$ is true, otherwise $I(A) = 0$.

**Step 3.** For $i = 1, 2, \cdots, n$, rearrange the point set in each dimension again by using the assigned probability:

$$\theta_{i,q}^{(2)} = F_{\Theta_i}^{-1}\left(\sum_{k=1}^{N} P_q \cdot I\left(\theta_{i,k}^{(1)} < \theta_{i,q}^{(1)}\right) + \frac{1}{2} \cdot P_q\right),\ q = 1, 2, \cdots, N \tag{B.3}$$

where the assigned probability $P_q$ is calculated by

$$P_q = \int_{\Omega_q} p_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \tag{B.4}$$

where $\Omega_q$ denotes the Voronoi region of the representative point $\boldsymbol{\theta}_q$ and satisfies

$$\Omega_q := \left\{\boldsymbol{\theta} \in \mathbb{R}^n : \left\|\boldsymbol{\theta} - \boldsymbol{\theta}_q\right\|_2 \leq \left\|\boldsymbol{\theta} - \boldsymbol{\theta}_j\right\|_2 \text{ for } j = 1, 2, \cdots, N\right\} \tag{B.5}$$

and $\|\cdot\|_2$ is the 2-norm.

By doing so, $\mathcal{P}_N^{(2)} = \{\boldsymbol{\theta}_q^{(2)} = (\theta_{1,q}^{(2)}, \theta_{2,q}^{(2)}, \cdots, \theta_{n,q}^{(2)})^\top\}_{q=1}^N$ is generated as the final point set with a smaller GF-discrepancy.

## Appendix C. Grassmann Gauss-Newton algorithm

Denote a many-to-one mapping $f : \boldsymbol{\theta} \in \mathbb{R}^n \mapsto x \in \mathbb{R}^1$ where $n \gg 1$. In ASM, the high-dimensional space $\mathbb{R}^n$ can be approximated from a relatively low-dimensional space $\mathbb{R}^m$ such that

$$f(\boldsymbol{\theta}) \approx g\left(\hat{\mathbf{W}}^\top \boldsymbol{\theta}\right), \text{ s.t. } \hat{\mathbf{W}} \in \mathbb{R}^{n\times m} \text{ and } \hat{\mathbf{W}}^\top \hat{\mathbf{W}} = \mathbf{I}_m \tag{C.1}$$

where $\mathbf{I}_m$ is an identity matrix of $m$ by $m$ for $1 \leq m \leq n$, and $g(\cdot)$ denotes the metamodel between $\hat{\mathbf{W}}^\top \boldsymbol{\theta} \in \mathbb{R}^m$ and $x \in \mathbb{R}^1$. Note that the orthogonal matrix $\hat{\mathbf{W}}$ is still unknown.

In the following part the top script of $\hat{\mathbf{W}}$ is omitted for simplicity without confusion.

Suppose we have $N$ samples $\{\boldsymbol{\theta}_k, x_k = f(\boldsymbol{\theta}_k)\}_{k=1}^N$, the gradient-free way to find $\mathbf{W}$ is to solve the following optimization issue, i.e.,

$$\underset{\mathbf{W}\in\mathbb{G}(m,\mathbb{R}^n)}{\arg\min} \sum_{k=1}^N \left[x_k - g\left(\mathbf{W}^\top \boldsymbol{\theta}_k\right)\right]^2 \tag{C.2}$$

where $\mathbb{G}(m, \mathbb{R}^n)$ denotes the Grassmann manifold of $m$-dimensional subspaces of $\mathbb{R}^n$, which can be regarded as the solution domain of $\mathbf{W}$.

Among many problems in solving Eq. (C.2), a good starting point $\mathbf{W}_0$ and an efficient iterative approach are of paramount significance. In this paper, three different starting points are generated individually by (1) random sampling method [36], (2) local linear regression method [23] and (3) Kriging-based method [30]. The first one is to randomly generate an orthogonal matrix $\mathbf{W}_0 \in \mathbb{R}^{n\times m}$ via the Gram-Schmidt orthonormalization, while the last two are to build a linear or Kriging model of $x = f(\boldsymbol{\theta})$, then we can compute an approximated $\mathbf{W}_0 \in \mathbb{R}^{n\times m}$ from Eqs. (17) to (18).

As for the iterative approach, the Grassmann Gauss-Newton method [36] is adopted in this paper. In [36], an approximated Gauss-Newton step for the optimization problem on the Grassmann manifold is proposed, where only the first derivative, i.e., the gradient is required. Moreover, an analytical Jacobian matrix is utilized to ensure a faster speed of convergence, based on the assumption that $g(\cdot)$ is of PCE type, i.e.,

$$g(\hat{\boldsymbol{\theta}}) := \sum_{\boldsymbol{a}\in\mathcal{A}} c_{\boldsymbol{a}} \Psi_{\boldsymbol{a}}(\hat{\boldsymbol{\theta}}), \ \hat{\boldsymbol{\theta}} = \mathbf{W}^\top \boldsymbol{\theta} \in \mathbb{R}^m \tag{C.3}$$

where $\mathcal{A}$ is the set of all orthogonal polynomials and $M = \text{card}(\mathcal{A})$, $\Psi_{\boldsymbol{a}}$ is generated in form of tensor production of one-dimensional and orthogonal polynomials, and $c_{\boldsymbol{a}}$ is the corresponding coefficients.

Denote $\mathbf{V}(\mathbf{W})$ be the information matrix whose generic term reads:

$$[\mathbf{V}(\mathbf{W})]_{i,j} := \Psi_j \left(\mathbf{W}^\top \boldsymbol{\theta}_i\right) \text{ for } i = 1, 2, \cdots, N \text{ and } j = 1, 2, \cdots, M \tag{C.4}$$

and let $\mathbf{c} = (c_1, c_2, \cdots, c_M)^\top \in \mathbb{R}^M$ be the coefficient vector of PCE, then Eq. (C.3) can be rewritten by

$$g\left(\mathbf{W}^\top \boldsymbol{\theta}\right) = \mathbf{V}\left(\mathbf{W}\right)\mathbf{c} \tag{C.5}$$

Let $\mathbf{f}$ be a vector of QoI, i.e., $\mathbf{f} = [f(\boldsymbol{\theta}_1), f(\boldsymbol{\theta}_2), \cdots, f(\boldsymbol{\theta}_N)]^\top \in \mathbb{R}^N$, then Eq. (C.2) can be rewritten by

$$\underset{\substack{\mathbf{c}\in\mathbb{R}^M \\ \mathbf{W}\in\mathbb{G}(m,\mathbb{R}^n)}}{\arg\min} \quad \|\mathbf{f} - \mathbf{V}(\mathbf{W})\mathbf{c}\|_2^2 \tag{C.6}$$

where we denote $\mathbf{r} = \mathbf{f} - \mathbf{V}(\mathbf{W})\mathbf{c}$ as the residual.

When the issue of optimization is in Euclidean space, the updated point $\mathbf{W}_+$ is searched from the present point $\mathbf{W}$ with a given direction $\boldsymbol{\Delta}$, i.e., $\mathbf{W}_+ = \mathbf{W} + \tau\boldsymbol{\Delta}$ where $\tau \in [0, 1]$ is a small number. While if the issue of optimization is on the Grassmann manifolds, the Newton's method is revised by [48]

$$\mathbf{W}_+ = \mathbf{W}\mathbf{D}\cos(\boldsymbol{\Sigma}\tau)\mathbf{D}^\top + \mathbf{S}\sin(\boldsymbol{\Sigma}\tau)\mathbf{D}^\top, \ \tau \in [0, 1] \tag{C.7}$$

where $\mathbf{S}\boldsymbol{\Sigma}\mathbf{D}^\top$ is the compact singular value decomposition of $\boldsymbol{\Delta}$.

In [36], the Gauss-Newton step $\boldsymbol{\Delta}$ is computed by an analytical Jacobian and the residual $\mathbf{r}$. The Grassmann Gauss-Newton algorithm is summarized in Algorithm 2.

**Algorithm 2** The Grassmann Gauss-Newton algorithm [36]

```
Require:
    Generate training samples, $\{\boldsymbol{\theta}_k, x_k = f(\boldsymbol{\theta}_k)\}_{k=1}^{N}$
    Determine the PCE structure, $g(\cdot)$
Ensure:
    PCE coefficients, $\mathbf{c} \in \mathbb{R}^M$
    Optimal active subspace, $\mathbf{W}_{\text{opt}} \in \mathbf{R}^{n\times m}$

Generate an initial point on $\mathbb{G}(m, \mathbb{R}^n)$, $\mathbf{W}_0 \in \mathbb{R}^{n\times m}$ s.t. $\mathbf{W}_0^\top \mathbf{W}_0 = \mathbf{I}_m$
Initial setup, $\mathbf{W} \leftarrow \mathbf{W}_0$
while not converged do
    Get active variable, $\hat{\boldsymbol{\theta}} \leftarrow \mathbf{W}^\top \boldsymbol{\theta}$
    Compute PCE coefficients, $\mathbf{c} \leftarrow \mathbf{V}(\mathbf{W})^+\mathbf{f}$        ▷ $^+$ is the Moore-Penrose pseudo-inverse
    Get the residual, $\mathbf{r} \leftarrow \mathbf{f} - \mathbf{V}\mathbf{c}$
    Compute the Gauss-Newton step, $\boldsymbol{\Delta}$ in [36]        ▷ require $\mathbf{W}$ and $\mathbf{r}$
    Update the current point, $\mathbf{W}_+ \leftarrow (\mathbf{W}, \boldsymbol{\Delta})$        ▷ Eq. (C.7)
    Get the new residual, $\mathbf{r}_+ \leftarrow \mathbf{f} - \mathbf{V}(\mathbf{W}_+)\mathbf{V}(\mathbf{W}_+)^+\mathbf{f}$
    $\mathbf{W} \leftarrow \mathbf{W}_+$
end while
```

### Acknowledgements

The Fundamental Research Funds for the Central Universities (Grant No. G2022KY05103) is sincerely appreciated by the first author. The Fundamental Research Funds for the Central Universities (Grant No. G2021KY05103) is gratefully appreciated by the corresponding author. The National Natural Science Foundation of China (Grant Nos. 72171194 and 51905430) as well as the Sino-German Mobility Programme (Grant No. M-0175) are highly appreciated by the fourth author, and the National Natural Science Foundation of China (Grant No. 51908468) is gratefully appreciated by the last author. The first author greatly thanks for constructive comments and discussions from the other authors. In particular, the constant support from the second author is greatly appreciated. As a younger researcher, the first author feels honoured to have supports from different institutes and groups, which inspires the first author with confidence in the future research.

## References

[1] H. Zhou, J. Li, X. D. Ren, Multi-scale stochastic structural analysis towards reliability assessment for large complex reinforced concrete structures, International Journal for Multiscale Computational Engineering 14 (3) (2016) 303–321. doi:10.1615/IntJMultCompEng.2016015745.

[2] V. Dubourg, B. Sudret, Meta-model-based importance sampling for reliability sensitivity analysis, Structural Safety 49 (2014) 27–36. doi:10.1016/j.strusafe.2013.08.010.

[3] Z. M. Jiang, J. Li, A new reliability method combining Kriging and probability density evolution method, International Journal of Structural Stability and Dynamics 17 (10) (2017) 1750113. doi:10.1142/S0219455417501139.

[4] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, Reliability Engineering and System Safety 93 (2008) 964–979. doi:10.1016/j.ress.2007.04.002.

[5] M. Moustapha, B. Sudret, Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework, Structural and Multidisciplinary Optimization 60 (2019) 2157–2176. doi:10.1007/s00158-019-02290-y.

[6] M. R. Rajashekhar, B. R. Ellingwood, A new look at the response surface approach, Structural Safety 12 (1993) 205–220. doi:10.1016/0167-4730(93)90003-J.

[7] C. G. Bucher, U. Bourgund, A fast and efficient response surface approach for structural reliability problems, Structural Safety 7 (1990) 57–66. doi:10.1016/0167-4730(90)90012-E.

[8] B. Echard, N. Gayton, M. Lemaire, AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation, Structural Safety 33 (2011) 145–154. doi:10.1016/j.strusafe.2011.01.002.

[9] J. Kim, J. Song, Probability-Adaptive Kriging in $n$-Ball (PAK-B$^{\mathrm{n}}$) for reliability analysis, Structural Safety 85 (2020) 101924. doi:10.1016/j.strusafe.2020.101924.

[10] R. Ghanem, P. D. Spanos, Polynomial chaos in stochastic finite elements, Journal of Applied Mechanics, Transactions of the American Society of Michanical Engineers 57 (1990) 197–202. doi:10.1115/1.2888303.

[11] C. Soize, R. Ghanem, Physical systems with random uncertainties: Chaos representations with arbitrary probability measure, SIAM Journal on Scientific Computing 26 (2) (2004) 395–410. doi:10.1137/S1064827503424505.

[12] J. Li, Stochastic Structural Systems: Analysis and Modeling (in Chinese), Science Press, Beijing, 1996.

[13] U. Alibrandi, A. Alani, C. G. Koh, Implications of high-dimensional geometry for structural reliability analysis and a novel linear response surface method based on SVM, International Journal of Computational Methods 12 (4) (2015) 1540016. doi:10.1142/S0219876215400162.

[14] J. Deng, D. S. Gu, X. B. Li, Z. Q. Yue, Structural reliability analysis for implicit performance functions using artificial neural network, Structural Safety 27 (2005) 25–48. doi:10.1016/j.strusafe.2004.03.004.

[15] L. Schueremans, D. Van Gemert, Benefit of splines and neural networks in simulation based structural reliability analysis, Structural Safety 27 (2005) 246–261. doi:10.1016/j.strusafe.2004.11.001.

[16] L. Comerford, I. A. Kougioumtzoglou, M. Beer, An artificial neural network approach for stochastic process power spectrum estimation subject to missing data, Structural Safety 52 (2015) 150–160. doi:10.1016/j.strusafe.2014.10.001.

[17] R. Ghanem, D. Higdon, H. Owhadi (Eds.), Handbook of Uncertainty Quantification, Springer International Publishing, Switzerland, 2017. doi:10.1007/978-3-319-12385-1.

[18] R. Alizadeh, J. K. Allen, F. Mistree, Managing computational complexity using surrogate models: a critical review, Research in Engineering Design 31 (2020) 275–298. doi:10.1007/s00163-020-00336-7.

[19] R. Teixeira, M. Nogal, A. O'Connor, Adaptive approaches in metamodel-based reliability analysis: A review, Structural Safety 89 (2021) 102019. doi:10.1016/j.strusafe.2020.102019.

[20] J. N. Fuhg, A. Fau, U. Nackenhorst, State-of-the-art and comparative review of adaptive sampling methods for Kriging, Archives of Computational Methods in Engineering 28 (2021) 2689–2747. doi:10.1007/s11831-020-09474-6.

[21] N. Lüthen, S. Marelli, B. Sudret, Sparse polynomial chaos expansions: Literature survey and benchmark, SIAM/ASA Journal on Uncertainty Quantification 9 (2) (2021) 593–649. doi:10.1137/20M1315774.

[22] T. Bui-Thanh, K. Willcox, O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, SIAM Journal on Scientific Computing 30 (6) (2008) 3270–3288. doi:10.1137/070694855.

[23] Z. M. Jiang, J. Li, High dimensional structural reliability with dimension reduction, Structural Safety 69 (2017) 35–46. doi:10.1016/j.strusafe.2017.07.007.

[24] I. M. Sobol', Theorems and examples on high dimensional model representation, Reliability Engineering and System Safety 79 (2003) 187–193. doi:10.1016/S0951-8320(02)00229-6.

[25] A. Nikishova, G. E. Comi, A. G. Hoekstra, Sensitivity analysis based dimension reduction of multiscale models, Mathematics and Computers in Simulation 170 (2020) 205–220. doi:10.1016/j.matcom.2019.10.013.

[26] M. Li, R. Q. Wang, G. F. Jia, Efficient dimension reduction and surrogate-based sensitivity analysis for expensive models with high-dimensional outputs, Reliability Engineering and System Safety 195 (2020) 106725. doi:10.1016/j.ress.2019.106725.

[27] G. Blatman, B. Sudret, Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach, Comptes Rendus Mécanique 336 (2008) 518–523. doi:10.1016/j.crme.2008.02.013.

[28] Y. N. Liu, M. Yousuff Hussaini, G. Ökten, Accurate construction of high dimensional model representation with applications to uncertainty quantification, Reliability Engineering and System Safety 152 (2016) 281–295. doi:10.1016/j.ress.2016.03.021.

[29] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis. The Primer, John Wiley & Sons, Chichester, 2008. doi:10.1002/9780470725184.

[30] P. G. Constantine, Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies, Society for Industrial and Applied Mathematics, Philadelphia, 2015. doi:10.1137/1.9781611973860.

[31] S. A. Billings, Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, John Wiley & Sons, New York, 2013. doi:10.1002/9781118535561.

[32] E. Jacquelin, N. Baldanzini, B. Bhattacharyya, D. Brizard, M. Pierini, Random dynamical system in time domain: A POD-PC model, Mechanical Systems and Signal Processing 133 (2019) 106251. doi:10.1016/j.ymssp.2019.106251.

[33] C. V. Mai, M. D. Spiridonakos, E. N. Chatzi, B. Sudret, Surrogate modelling for stochastic dynamical systems by combining NARX models and polynomial chaos expansions, International Journal for Uncertainty Quantification 6 (4) (2016) 313–339. doi:10.1615/Int.J.UncertaintyQuantification.2016016603.

[34] R. Schöbi, B. Sudret, J. Wiart, Polynomial-chaos-based Kriging, International Journal for Uncertainty Quantification 5 (2) (2015) 171–193. doi:10.1615/Int.J.UncertaintyQuantification.2015012467.

[35] J. B. Chen, J. Y. Yang, J. Li, A GF-discrepancy for point selection in stochastic seismic response analysis of structures with uncertain parameters, Structural Safety 59 (2016) 20–31. doi:10.1016/j.strusafe.2015.11.001.

[36] J. M. Hokanson, P. G. Constantine, Data-driven polynomial ridge approximation using variable projection, SIAM Journal on Scientific Computing 40 (3) (2018) A1566–A1589. doi:10.1137/17M1117690.

[37] J. Li, J. B. Chen, Stochastic Dynamics of Structures, John Wiley & Sons, Singapore, 2009. doi:10.1002/9780470824269.

[38] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, The Annals of Statistics 32 (2) (2004) 407–499. doi:10.1109/glocom.1988.25879.

[39] G. Berkooz, P. Holmes, J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annual Review of Fluid Mechanics 25 (1993) 539–575. doi:10.1146/annurev.fl.25.010193.002543.

[40] T. M. Russi, Uncertainty Quantification with Experimental Data and Complex System Models, Phd thesis, Dissertation submitted to University of California, Berkeley (2010).

[41] P. G. Constantine, E. Dow, Q. Q. Wang, Active subspace methods in theory and practice: Applications to Kriging surfaces,

SIAM Journal on Scientific Computing 36 (4) (2014) A1500–A1524. doi:10.1137/130916138.

[42] K. D. Coleman, A. Lewis, R. C. Smith, B. Williams, M. Morris, B. Khuwaileh, Gradient-free construction of active subspaces for dimension reduction in complex models with applications to neutronics, SIAM/ASA Journal on Uncertainty Quantification 7 (1) (2019) 117–142. doi:10.1137/16M1075119.

[43] B. F. Feeny, R. Kappagantu, On the physical interpretation of proper orthogonal modes in vibrations, Journal of Sound and Vibration 211 (4) (1998) 607–616. doi:10.1006/jsvi.1997.1386.

[44] J. B. Chen, J. P. Chan, Error estimate of point selection in uncertainty quantification of nonlinear structures involving multiple nonuniformly distributed parameters, International Journal for Numerical Methods in Engineering 118 (2019) 536–560. doi:10.1002/nme.6025.

[45] G. Kewlani, J. Crawford, K. Iagnemma, A polynomial chaos approach to the analysis of vehicle dynamics under uncertainty, Vehicle System Dynamics 50 (5) (2012) 749–774. doi:10.1080/00423114.2011.639897.

[46] L. G. Crespo, S. P. Kenny, The NASA langley challenge on optimization under uncertainty, Mechanical Systems and Signal Processing 152 (2021) 107405. doi:10.1016/j.ymssp.2020.107405.

[47] R. Schöbi, S. Marelli, B. Sudret, UQLab user mannual - Polynomial chaos Kriging. UQLab-V1.3-109, Tech. rep., Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland (2019).

[48] A. Edelman, T. A. Arias, S. T. Smith, The geometry of algorithms with orthogonality constraints, SIAM Journal on Matrix Analysis and Applications 20 (2) (1998) 303–353. doi:10.1137/S0895479895290954.