Fast Prediction of Dynamic Structural Response Using Reduced Basis Function Combined with Neural Network

JIXING CAO, SER-TONG QUEK, SHANLI ZHANG,
CHI ZHANG, MINBO CAI and MICHAEL SI

ABSTRACT

Frequent calibration of the dynamic characteristics of jack-up platforms are important for accurate evaluation of their safety and comfort to occupants. As jack-ups are large in scale involving many physical mechanisms and hydrodynamic coupling, it is challenging to rapidly optimize the estimation of system parameters and predict the dynamic responses under potential scenarios. This study develops a novel offline-online framework to address these issues. In the offline phase, a set of reduced basis functions is extracted from a collection of high-fidelity datasets, and the corresponding coefficients are employed to train neural network models. The online phase involves mapping model parameters to the coefficients of reduced basis functions using the trained neural network. The trained neural network combined with the reduced basis function is then utilized to predict dynamic response. The feasibility of the proposed method was evaluated through a numerical model of a jack-up under wave loads. The results indicate that the method is effective, robust, and promising for the rapid evaluation of large-scale structures.

1 INTRODUCTION

Jixing Cao, Department of Civil & Environmental Engineering, National University of Singapore, Singapore 117576, Singapore
Ser-Tong Quek, Department of Civil & Environmental Engineering, National University of Singapore, Singapore 117576, Singapore
Shanli Zhang, Technology Centre for Offshore and Marine, Singapore (TCOMS) 118411, Singapore
Chi Zhang, Technology Centre for Offshore and Marine, Singapore (TCOMS) 118411, Singapore
Minbo Cai, Technology Centre for Offshore and Marine, Singapore (TCOMS) 118411, Singapore
Michael Si, Department of Civil & Environmental Engineering, National University of Singapore, Singapore 117576, Singapore; Technology Centre for Offshore and Marine, Singapore (TCOMS) 118411, Singapore

Physics-based modeling and simulation have been widely applied in the field of offshore structures to compute their dynamic response and associated internal forces under diverse loading scenarios. Achieving high-fidelity solutions for offshore structures entails substantial computational burden [1] and becomes a challenge when speedy simulations and/or numerous repetitions are necessary. For example, real-time calculation is indispensable in predictive control, while structural reliability analysis demands thousands of repetitive computations. Therefore, it is imperative to develop a numerical model that can mitigate computational demand with minimal compromise in accuracy.

Some studies focused on developing the surrogate model to replace the original high-fidelity model to achieve fast computation with good accuracy [2]. Such models employ mathematical equations or algorithms to map the input-output relationship of a given system. The most straightforward method to regress the input-output data is polynomial least-squares. However, its application is limited to simple systems. Another prevalent class of technique is projection-based model reduction, such as proper orthogonal decomposition (POD) and singular value decomposition (SVD), which approximate high-dimensional dynamic systems by employing a low-dimensional subspace. It has been successfully applied in the domain of fluid mechanics such as flow-field problems [3,4], and proven to be a useful technique for extracting salient features in complex systems.

Recently, artificial neural network (NN) shows promising capability to capture the underlying nonlinear input–output relationship for fluid mechanics problems. Lee and Carlberg [5] developed a novel framework of deep convolutional autoencoder to reduce the dynamic systems into nonlinear manifolds. Kutyniok et al. [6] established upper bounds using Rectified Linear Unit (ReLU) NN to approximate the solution maps of complex partial differential equations. However, the NN model serves as black-box and lacks generalization beyond the training data employed.

This work presents a model order reduction framework that combines the Reduced Basis and Neural Network (RB-NN) for time-dependent systems. A two-step POD algorithm is adopted to reduce the high-dimensional data matrices. An offline-online procedure is developed to facilitate real-time prediction, where the reduced order NN model is trained offline, and the field prediction using the trained model is performed in real time. A jack-up model subjected to wave loads is analyzed to demonstrate the accuracy and robustness of the proposed framework.

2 FRAMEWORK OF RB-NN

2.1 Reduced basis function using proper orthogonal decomposition

The general equation for discrete dynamical systems under wave loading is described by

$$\frac{d}{dt}\boldsymbol{x}(t) = f(\boldsymbol{x}, \boldsymbol{u}, t) \tag{1}$$

in which f represents the nonlinear system, $\boldsymbol{x}$ is the state vector, t is the time step, and $\boldsymbol{u}$ is the external force.

To obtain the structural response, Equation (1) can be integrated using a time-stepping algorithm,

$$\boldsymbol{x}(t+\Delta t)=\boldsymbol{x}(t)+\int_t^{t+\Delta t} f(\boldsymbol{x},\boldsymbol{u},\tau)d\tau \tag{2}$$

where Δt is the chosen time interval.

Based on Equation (2), the high-fidelity structural response at all the number of degree of freedoms N_d at time step t for a specified realization of the parameters $\boldsymbol{\mathcal{P}}^{(k)}$ can be computed. The responses at time $t_1, t_2, \cdots, t_{\mathrm{N_t}}$ can be put in matrix form as

$$\mathbf{X}^{(\boldsymbol{\mathcal{P}}^{(k)})}=\begin{bmatrix} x(N_1,t_1) & x(N_1,t_2) & \cdots & x(N_1,t_{\mathrm{N_t}}) \\ x(N_2,t_1) & x(N_2,t_2) & \cdots & x(N_2,t_{\mathrm{N_t}}) \\ \vdots & & \ddots & \vdots \\ x(N_d,t_1) & x(N_d,t_2) & \cdots & x(N_d,t_{\mathrm{N_t}}) \end{bmatrix} \tag{3}$$

where $\mathbf{X}^{(\boldsymbol{\mathcal{P}}^{(k)})} \in \mathbb{R}^{\mathrm{N_d}\times\mathrm{N_t}}$, $\boldsymbol{\mathcal{P}}=\{\boldsymbol{\mathcal{P}}^{(1)},\cdots,\boldsymbol{\mathcal{P}}^{(k)},\cdots,\boldsymbol{\mathcal{P}}^{(\mathrm{N_P})}\} \in \mathbb{R}^{\mathrm{N_V}\times\mathrm{N_P}}$ denotes parameter space, $\mathrm{N_V}$ is the number of parameters, $\mathrm{N_P}$ is the total number of realizations. $\mathbf{X}^{(\boldsymbol{\mathcal{P}}^{(k)})}$ can be rearranged into a column vector $\mathbf{Y}^{(\mathrm{k})}$ of size $\mathrm{N_d N_t}\times 1$. A collection of high-fidelity solution for all parameters is denoted by

$$\mathbf{Y}=[\mathbf{Y}^{(1)},\mathbf{Y}^{(2)},\cdots,\mathbf{Y}^{(\mathrm{N_P})}] \in \mathbb{R}^{\mathrm{N_d N_t}\times\mathrm{N_P}} \tag{4}$$

Conceptually, $\mathbf{Y}$ can be reduced by projecting onto a low-dimensional space based on SVD

$$\mathbf{Y}=\mathbf{U\Sigma V}^{\boldsymbol{T}} \tag{5}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, $\boldsymbol{\Sigma}=\mathrm{diag}[\sigma_1,\cdots,\sigma_A,0,\cdots,0]$ is a diagonal matrix with singular values, and subscript A is the rank of $\mathbf{Y}$. The matrix of reduced basis vector is denoted as $\boldsymbol{\Phi}_R=[\boldsymbol{\psi}_1,\cdots,\boldsymbol{\psi}_R] \in \mathbb{R}^{\mathrm{N_d N_t}\times\mathrm{R}}$, where R is the number of reduced basis (also known as the number of POD modes). R can be determined by

$$1-\frac{\sum_{i=1}^{R}\sigma_i^2}{\sum_{i=1}^{A}\sigma_i^2} \le \epsilon_{\mathrm{POD}} \tag{6}$$

where ϵ_{POD} is the tolerable relative error. Using $\boldsymbol{\Phi}_R$, $\mathbf{Y}$ can be reformulated as

$$\mathbf{Y}=\boldsymbol{\Phi}_R\boldsymbol{\alpha}_{\boldsymbol{Y}} \tag{7}$$

where the associated coefficients, $\boldsymbol{\alpha}_{\boldsymbol{Y}} \in \mathbb{R}^{\mathrm{R}\times\mathrm{N_t}}$, can be obtained from $\boldsymbol{\Phi}_R$ and $\mathbf{Y}$ through the pseudo-inverse.

For large-scale structures with large number of time steps considered, the dimension of $\mathbf{Y}$ is so large that the POD algorithm may not be directly applied to decompose the data because the operation of SVD for large matrix is expensive. To overcome this, a two-step POD algorithm is used. First, POD is performed on the response for each realization $\mathbf{X}^{(\boldsymbol{\mathcal{P}}^{(k)})}$ (see Equation (3)), from which a matrix of r reduced basis vectors $\boldsymbol{\chi}_{\mathrm{r}}^{\boldsymbol{\mathcal{P}}^{(k)}}$ can be obtained. In order to have a common set of reduced vectors for all realizations, $\boldsymbol{\chi}_{\mathrm{r}}^{\boldsymbol{\mathcal{P}}^{(k)}}$ for all k can be assembled as $\mathbb{T} = \left[\boldsymbol{\chi}_{\mathrm{r}}^{\boldsymbol{\mathcal{P}}^{(1)}}, \boldsymbol{\chi}_{\mathrm{r}}^{\boldsymbol{\mathcal{P}}^{(2)}}, \cdots, \boldsymbol{\chi}_{\mathrm{r}}^{\boldsymbol{\mathcal{P}}^{(k)}}\right]$ to perform another SVD operation. This will yield one common reduced space, represented by $\boldsymbol{\Phi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 \cdots, \boldsymbol{\psi}_R] \in \mathbb{R}^{\mathrm{N_d} \times \mathrm{R}}$ where R is the number of reduced basis for $\mathbb{T}$. The associated coefficients matrix $\boldsymbol{\alpha}_k$ for each k can be determined via pseudo-inversion.

Hence, the above will yield $\mathrm{N_P}$ sets of $(\boldsymbol{\mathcal{P}}^{(k)}, \boldsymbol{\alpha}_k)$ which will form as input and output pair for training the NN model

2.2 Neural network model to estimate reduced coefficients

The objective of the NN model is to approximate a mapping function between the input parameters $\boldsymbol{\mathcal{P}}$ and the output reduced coefficients $\boldsymbol{\alpha}$. A NN architecture is constructed with multiple layers that iteratively tune the NN parameters (i.e., weights and biases) to approximate the underlying mapping function. Specifically, the outputs of the l-th layer $\boldsymbol{Z}^l$ is related to the outputs of the $(l-1)$-th layer $\boldsymbol{Z}^{l-1}$ by

$$\boldsymbol{Z}^l = \zeta(\boldsymbol{W}^l \boldsymbol{Z}^{l-1} + \boldsymbol{b}^l) \tag{8}$$

where $\boldsymbol{W}^l$ and $\boldsymbol{b}^l$ are the weight matrix and bias parameters for the l-th layer; and ζ is the activation function. In this study, the ReLU activation function is utilized for all hidden layers and a linear activation function is employed for the output layer. The ReLU function, with χ as the input to the node, can be mathematically expressed as

$$\zeta(\chi) = \max(0, \chi) \tag{9}$$

For NN with L number of hidden layers between the input and output layers, the resulting mapping function can be formulated as

$$\boldsymbol{\alpha} = \zeta_L\left(\boldsymbol{W}^L, \boldsymbol{b}^L, \cdots, \zeta_2\left(\boldsymbol{W}^2, \boldsymbol{b}^2, \zeta_1(\boldsymbol{W}^1, \boldsymbol{b}^1, \boldsymbol{\mathcal{P}})\right)\right) \tag{10}$$

where $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\alpha}$ are the input and output of the NN, respectively.

The goal of the training process is to find suitable values of the parameters $\boldsymbol{W}^l$ and $\boldsymbol{b}^l$ for $l = 1, 2, \cdots, L$. This is accomplished through minimization of the loss function, where the mean square error (MSE) between the true data $\boldsymbol{\alpha}^{(\mathrm{T})}$ and the output of the NN $\boldsymbol{\alpha}(\boldsymbol{\mathcal{P}})$ is defined as

$$\boldsymbol{C} = \frac{1}{N_s} \sum_{i=1}^{N_s} \left(\boldsymbol{\alpha}(\boldsymbol{\mathcal{P}}) - \boldsymbol{\alpha}^{(\mathrm{T})}\right)^2 \tag{11}$$

where $\boldsymbol{C}$ is the loss function and N_s is the number of samples. The ADAM (Adaptive Moment Estimation) optimization algorithm is employed to minimize the loss function. The process involves the gradients of the loss function with respect to the model's parameters, which are computed using the backpropagation method.

Figure 1 depicts the proposed RB-NN method, which comprises four distinct stages: (a) collection of high-fidelity data and the construction of a set of reduced basis; (b)

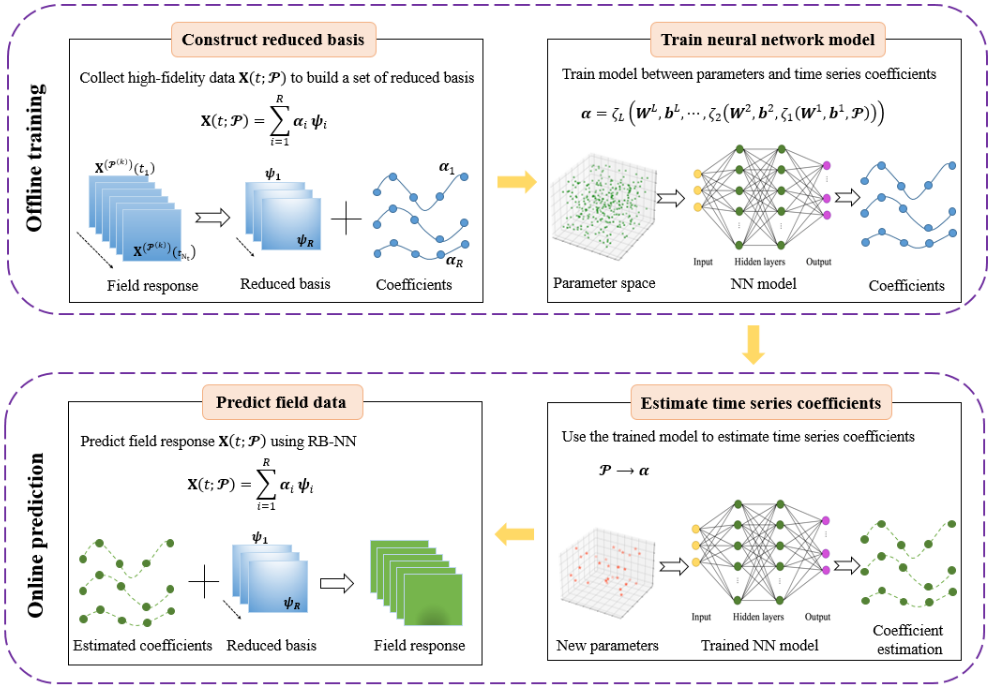

Figure 1. RB-NN framework.

training of a NN model to tune the time series coefficients; (c) prediction of time series coefficients for new parameters using the trained model; and (d) prediction of field response at any given time via RB-NN. The stages (a) and (b) necessitate computationally demanding tasks to construct the low-dimensional subspace and train the coefficients of reduced basis arrays using NNs. These tasks are performed only once during the offline phase. In contrast, the online stage comprises stages (c) and (d), which enable efficient computation using the RB-NN model approximation for any new set of parameter values.

3 REDUCED-ORDER MODEL OF JACK-UP PLATFORM

3.1 Numerical model and data collection

This section presents the development of a numerical model for performing hydrodynamic analysis of jack-up platform, to illustrate the proposed method. Figure 2 shows the finite element model of the jack-up platform comprising a hull and three vertical legs. The hull was assumed to be a rigid body and modelled with some nodes connected by weightless rigid elements. To simulate the drilling units, a cantilever beam located at the aft end of the hull was modelled. The legs, on the other hand, were modelled using triangulated frame elements. Since the loading from the hull was transmitted to the legs through guides, spring elements were assigned between the legs and the hull.

In order to account for the foundation fixity, spring elements were affixed between the legs and seafloor which enabled computing the overturning resistance provided by the spudcan. Additionally, the wave model was built based on the fifth-order Stokes

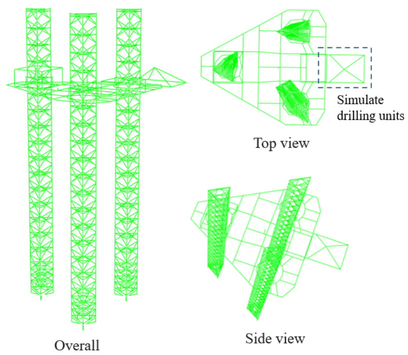

Figure 2. Finite element model of jack-up platform developed in ABAQUS.

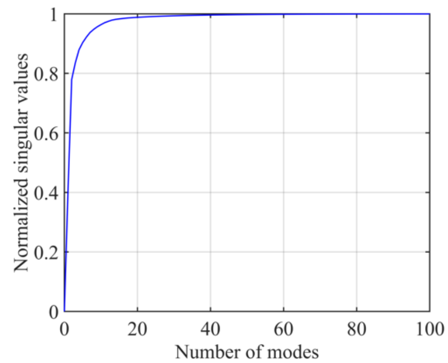

Figure 3. Cumulative normalized singular values of first 100 modes.

wave theory [7]. ABAQUS software [8] provided a predefined function for modeling Stokes fifth-order waves, which required three parameters: wave height (H_{w}), period of wave (τ_w) and phase angle (θ_w). The time history of the structural response was computed using the explicit time-stepping procedure.

The aforementioned full-order model served as a data generator for producing both the training and validation dataset. Specifically, two input variables, namely the wave height (H_w) and the phase angle (θ_w), were randomly sampled within the uniformly distributed parameter space $\boldsymbol{\mathcal{P}} = \{H_w \in [0, 0.62], \theta_w \in [-1,1]\}$, yielding 300 pairs of realizations. These sets were then utilized as inputs for the full-order model, enabling the computation of the responses of interest, which in this study is the time history of displacement for all nodes. A total of 300 input-output pairs of synthetic data were collected for training and validating the RB-NN model.

3.2 RB-NN model training and validation

The 300 sets of data were randomly split into 240 sets for training the NN and 60 sets for validation. To improve the stability and performance, the data were normalized. The training data was used to build the reduced basis in the POD phase. Figure 3 shows the cumulative normalized singular values with the number of modes. The increase in the cumulative normalized singular values was steep for the initial ten modes, after which a gradual asymptotic behavior towards 1 was observed. With a POD tolerance ϵ_{POD} of 10^{-4}, the first 91 POD modes were chosen to construct the reduced basis.

The coefficients of POD modes extracted from the high-fidelity datasets were utilized for training the NN model. The NN comprised a solitary input layer, four hidden layers, each of which had 32 neurons, and a single output layer. The training of the NN was accomplished through minimization of the loss function in Equation (11), utilizing the gradient descent method, over 100 epochs. Figure 4 depicts the decay of the MSE value with respect to the number of epochs. The MSE values exhibited significant reduction during the initial 15 epochs, and subsequently reached a plateau around 100 epochs, indicating that under-fitting did not adversely impact the training process. Furthermore, the MSE for both the training and validation datasets displayed a similar trend, implying that overfitting also did not occur.

Figure 5 compares the outputs of the NN with the corresponding targets. The dashed line in each subplot represented the network is exactly equal to the target, while the solid

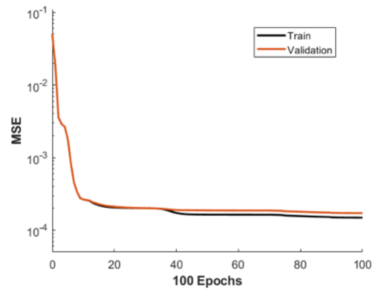

Figure 4. Trend of MSE for train and validation datasets.

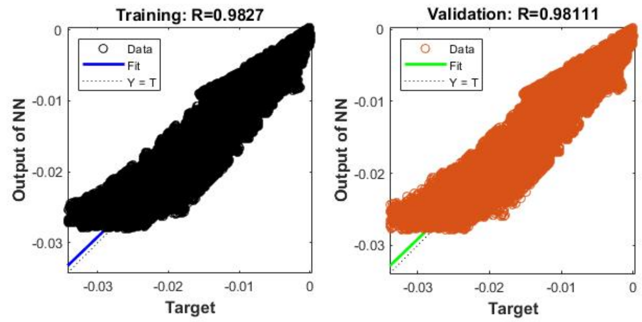

Figure 5 Regression between outputs of NN and targets.

line represented the best fit linear regression between outputs and targets. The R values obtained for the training and validation datasets were both greater than 0.98, indicating that the NN outputs closely align with the targets.

Different reduced basis may also influence the accuracy of the RB-NN model. The tolerances for POD were set at $\epsilon_{\text{POD}} = 10^{-1}$, $\epsilon_{\text{POD}} = 10^{-2}$, $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{POD}} = 10^{-4}$, resulting in the corresponding number of POD modes as $R = 7$, $R = 21$, $R = 55$ and $R = 91$, respectively. Upon successful training and validation of the four RB-NN models, new parameters were inputted to generate the dynamic responses. The relative error (τ) was utilized to evaluate the dissimilarities between the dynamic responses generated by the RB-NN model and the full-order model, given by

$$\boldsymbol{\tau} = \frac{\boldsymbol{\alpha}(\boldsymbol{\mathcal{P}}) - \boldsymbol{\alpha}^{(\mathrm{T})}}{\boldsymbol{\alpha}^{(\mathrm{T})}}, \tag{12}$$

where $\boldsymbol{\alpha}^{(\mathrm{T})}$ is the true data and $\boldsymbol{\alpha}(\boldsymbol{\mathcal{P}})$ is the output of the NN.

Figure 6 displays the box plots of relative errors for different numbers of POD modes employed where the bottom, central and top line of the green box represents the 25$^{\text{th}}$, 50$^{\text{th}}$ and 75$^{\text{th}}$ percentile errors, respectively, and the 'whiskers' represent the minimum and maximum errors. The results showed improvement in prediction accuracy with decreased median error as the number of modes increased from 7 to 91.

The computation times for the four RB-NN models are summarized in Table 1, based on a desktop equipped with Intel(R) Core(TM) i5-7400 @ 3.00 GHz CPU and an NVIDIA GeForce GT 730 GPU. The training time for NN was substantially greater than that for prediction. The training time increases exponentially as the number of modes increases, while the prediction time exhibited a linear increase. This trend illustrates the tradeoff between model accuracy and computational efficiency.

4 CONCLUSION

This paper presents a reduced-order-cum-NN model framework for dynamic response simulation of offshore structures. A two-step procedure was employed to efficiently extract the POD modes from data matrices to yield the reduced basis with specific tolerance error. The coefficients of POD modes were used to train and validate the NN model. Given the computationally intensive nature of the process, it was

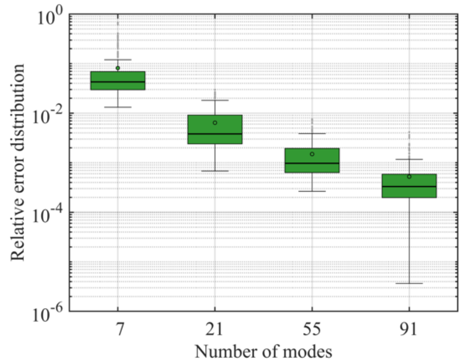

Figure 6. Relative error for different number of modes used.

TABLE 1 COMPARISON OF COMPUTING TIME (UNIT: SECOND).

Number of modes	Training	Prediction
7	1854	1.30E-03
21	4044	2.10E-03
55	8442	3.80E-03
91	21006	5.60E-03

executed offline. In the online phase, structural response can be predicted using the trained model with the given parameters.

The offline-online framework was applied to a jack-up platform for the prediction of dynamic response. The results demonstrate that a well-trained surrogate model can accurately predict the structural response in almost real time. This method offers the advantage of constructing a surrogate model using a relatively small dataset. The constructed surrogate model can be further employed for parameter optimization and reliability analysis.

ACKNOWLEDGEMENTS

This work is funded by the Enhancing Offshore System Productivity, Integrity and Survivability in Extreme Environments (ENSURE) programme. The fund, supported by A*STAR under its RIE 2020 Industry Alignment Fund (Grant No: A19F1a0104), is gratefully acknowledged.

REFERENCES

1. Siddiqui, M. S., Latif, S. T. M., Saeed, M., Rahman, M., Badar, A. W., & Hasan, S. M. 2020. “Reduced order model of offshore wind turbine wake by proper orthogonal decomposition,” International Journal of Heat and Fluid Flow, 82, 108554.
2. Brunton, S. L., Noack, B. R., & Koumoutsakos, P. 2020. “Machine learning for fluid mechanics,” Annual review of fluid mechanics, 52, 477-508.
3. Weller, J., Lombardi, E., Bergmann, M., & Iollo, A. 2010. “Numerical methods for low‐order modeling of fluid flows based on POD,” International Journal for Numerical Methods in Fluids, 63(2), 249-268.
4. Liberge, E., Benaouicha, M., & Hamdouni, A. 2007. “Proper orthogonal decomposition investigation in fluid structure interaction,” European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique, 16(3-4), 401-418.
5. Lee, K., & Carlberg, K. T. 2020. “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders,” Journal of Computational Physics, 404, 108973.
6. Kutyniok, G., Petersen, P., Raslan, M., & Schneider, R. 2022. “A theoretical analysis of deep neural networks and parametric PDEs,” Constructive Approximation, 55(1), 73-125.
7. Fenton, J. D. 1985. A fifth-order Stokes theory for steady waves. Journal of waterway, port, coastal, and ocean engineering, 111(2), 216-234.
8. Abaqus G. Abaqus 6.11. 2011. Dassault Systemes Simulia Corporation, Providence, RI, USA.