

# Deep Learning-Based Model Reduction for Distributed Parameter Systems

Mingliang Wang, *Member, IEEE*, Han-Xiong Li, *Fellow, IEEE*, Xin Chen, and Yun Chen

**Abstract**—This paper presents a deep learning-based model reduction method for distributed parameter systems (DPSs). The proposed method includes three phases. In phase I, numerical or experimental data of the spatiotemporal distribution is reduced into low-dimensional representations using the deep auto-encoder (DAE). In phase II, the low-dimensional representations are used to establish the reduced-order model. In phase III, the reduced model is then used to reconstruct the high-dimensional DPS. Experimental studies are conducted to validate the proposed method. The proposed method is compared with the classical proper orthogonal decomposition method and demonstrates better modeling accuracy and efficiency in the experiments.

**Index Terms**—Deep learning, distributed parameter system (DPS), model reduction, restricted Boltzmann machine (RBM), spatiotemporal dynamics.

## I. INTRODUCTION

MANY spatiotemporal dynamics process, such as thermal process, biological process, and chemical reaction processes are usually described by partial differential equations (PDEs). Analytical solutions to these distributed parameter systems (DPSs) are hard to obtain [1]. Numerical methods, such as finite difference method [2] and finite element method (FEM) [3], are usually applied to solve these equations for the simulation and design purpose. However, the numerical solution of these equations are computationally costly and impractical for real-time applications such as monitoring [4] and control [5].

Model reduction methods are then developed to capture the main dynamics of the process, through which a computational-efficient approximation solution can be obtained [6], [7]. The most widely used model reduction methods are the spectral method [8] and proper orthogonal decomposition (POD) method [9]. They both write the solution in space-time separation form [10]. The spectral method can achieve the

Manuscript received April 28, 2016; revised July 2, 2016; accepted August 9, 2016. Date of publication September 21, 2016; date of current version November 15, 2016. This work was supported in part by the Projects from Research Grants Council of Hong Kong under Grant 11205615 and Grant 11207714 and in part by the Project from Guangdong Government under Grant 2016A010102016. This paper was recommended by Associate Editor J.-H. Chou.

M. Wang and H.-X. Li are with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong (e-mail: minglwang2-c@my.cityu.edu.hk; mehxli@cityu.edu.hk).

X. Chen and Y. Chen are with the Key Laboratory of Mechanical Equipment Manufacturing and Control Technology of Ministry of Education, Guangdong University of Technology, Guangzhou 510006, China.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMC.2016.2605159

lower-order model than the FEM by truncating the fast modes from the spatial frequency domain [11], [12]. However, the spectral method has many limitations. For instance, it needs the PDEs and parameters to be known explicitly. It also requires the process that has a regular space domain and smooth output [10], [13]. Analogous to spectral method, the POD method reduces the model by finding dominant modes of the system. However, the difference is that the POD method identifies the empirical eigenfunctions from the numerical or experimental data. This data driven approach allows the POD method can be applied to the unknown DPS and complex geometry [14]. The POD method typically works on the high dissipative systems. But the performance of the POD method deteriorates when dealing with hyperbolic dynamics such as traveling hump according to [15].

In the field of machine learning, deep learning is a popular concept which aims to learn the high-level abstractions of data [16]. Typically, the deep learning structure are built on restricted Boltzmann machines (RBMs) [17]. Many deep learning methods have been developed successfully in various applications, such as deep convolutional learning in computer vision [18] and deep multimodal learning in speech recognition [19]. Among them, DAE is designed for nonlinear dimensionality reduction [20]. The structure of DAE is similar to the auto-associative neural networks [21]. However, the DAE adopts the energy-based stochastic gradient method [22] as pretraining approach and then “unfolded” adjacent layers to conduct the back-propagation algorithm [23] for fine-tuning of parameters. With the pretraining, the method can efficiently avoid the local minimum problems [24].

In this paper, a deep learning-based model reduction method is developed for a class of DPSs such as advection problem. The classic POD method works poorly on these systems. The proposed method includes three phases (I, II, III). In phase I, numerical or experimental data of the output is reduced into low-dimensional representations using the DAE. In phase II, low-dimensional representations are used to establish the reduced-order method which models the low-dimensional temporal dynamics. In phase III, the approximated representations are then used to reconstruct the high-dimensional output. Experiments are conducted to validate the proposed method in different DPS. The proposed method is a data-based low-order modeling approach using the neural network-based RBMs and DAEs. Due to the deep learning structure, it can achieve better modeling performance than the classical POD when applied to the complex DPS, especially the nonparabolic systems.

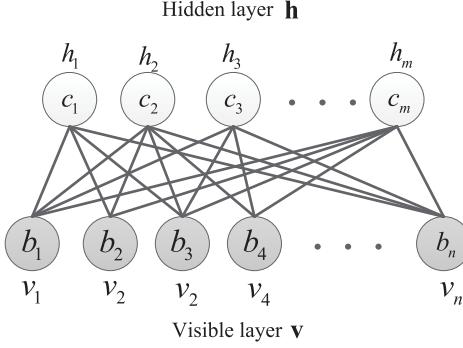


Fig. 1. Undirected graphical model of RBM.

The rest of this paper is organized as follows. In Section II, the RBM and DAE are briefly introduced. The proposed modeling approach is demonstrated in Section III. Simulation examples illustrating the performance of the proposed modeling approach are presented in Section IV. The discussion of advantages of the proposed method is conducted in Section V. And finally, some conclusions are provided in Section VI.

## II. RESTRICTED BOLTZMANN MACHINE AND DEEP AUTO-ENCODER

### A. Restricted Boltzmann Machine

The RBM is an energy-based undirect graphical model [25], [26] as shown in Fig. 1. The energy function of the model is defined as follows:

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i=1}^n b_i v_i - \sum_{j=1}^m c_j h_j - \sum_{i=1}^n \sum_{j=1}^m w_{ij} v_i h_j \quad (1)$$

where  $v_i$  and  $h_j$  are visible and hidden units in binary states  $\{0, 1\}$ ,  $w_{ij}$  is the real valued connection weights, and  $b_i$  and  $c_j$  are their real valued bias terms. With the energy defined, the joint probability distribution of the graphical model are Gibbs distribution given as

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})} \quad (2)$$

where  $Z = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}$  is the partition function. Based on the definitions in (1) and (2)

$$p(\mathbf{h}|\mathbf{v}) = \frac{p(\mathbf{v}, \mathbf{h})}{p(\mathbf{v})} = \frac{\frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})}}{\frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}}. \quad (3)$$

With the conditional independence, the conditional property can be also written as

$$p(\mathbf{h}|\mathbf{v}) = \prod_j^m p(h_j|\mathbf{v}). \quad (4)$$

With (3) and (4) and the binary property of the variables  $\mathbf{h}$ , the conditional probability of a single variable can be derived as (5a)

$$p(h_j = 1|\mathbf{v}) = \frac{1}{1 + e^{-(\sum_i^n w_{ij} v_i + c_j)}} \quad (5a)$$

$$p(v_i = 1|\mathbf{h}) = \frac{1}{1 + e^{-(\sum_j^m w_{ij} h_j + b_i)}}. \quad (5b)$$

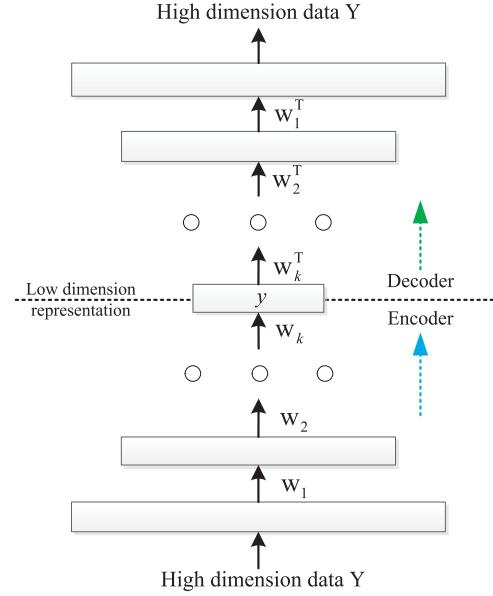


Fig. 2. Structure of DAE.

Similar results can be derived for the conditional probability of single visible variable  $v_i$  as (5b). The RBM is indeed an Markov random field (MRF). The log-likelihood gradient of MRFs with latent variables can be derived as follows:

$$\begin{aligned} \frac{\partial \ell(\boldsymbol{\theta}|\mathbf{v})}{\partial \boldsymbol{\theta}} &= \frac{\ln p(\mathbf{v}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\ln \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}}{\partial \boldsymbol{\theta}} \\ &= - \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}) \frac{\partial E(\mathbf{v}|\mathbf{h})}{\partial \boldsymbol{\theta}} + \sum_{\mathbf{v}, \mathbf{h}} \frac{\partial E(\mathbf{v}|\mathbf{h})}{\partial \boldsymbol{\theta}} \end{aligned} \quad (6)$$

where  $\boldsymbol{\theta}$  is the model parameters including  $w_{ij}$ ,  $b_i$ , and  $c_j$ . For a training set  $\mathbf{V} = \{v_1, v_2, \dots, v_k\}$ , the mean for the log-likelihood gradient of  $w_{ij}$  can be written as

$$\begin{aligned} \frac{1}{k} \sum_{\mathbf{v} \in \mathbf{V}} \frac{\partial \ell(w_{ij}|\mathbf{v})}{\partial w_{ij}} &= \frac{1}{k} \sum_{\mathbf{v} \in \mathbf{V}} [\mathbb{E}_{p(\mathbf{h}|\mathbf{v})}[v_i h_j] + \mathbb{E}_{p(\mathbf{h}, \mathbf{v})}[v_i h_j]] \\ &\propto \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \end{aligned} \quad (7)$$

where  $v_i h_j$  is derived by  $[\partial E(\mathbf{v}, \mathbf{h}) / \partial w_{ij}]$ ,  $\langle \rangle$  denotes the statistical expectation. Analogously, the log-likelihood gradients for  $b_i$  and  $c_j$  can be derived. To obtain the unbias estimation of the log-likelihood gradients in (7), Markov chain Monte Carlo (MCMC) methods [27] are typical used, which requires enormous sampling steps. However, the contrastive divergence learning [28] shows that the Gibbs chain (MCMC) [29] run for only  $z$  steps (usually set to 1) is sufficient for the parameter training. Then the updating rule can be formulated as

$$\begin{aligned} \frac{\partial \ell(\boldsymbol{\theta}|\mathbf{v}^{(0)})}{\partial \boldsymbol{\theta}} &= - \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^{(0)}) \frac{\partial E(\mathbf{v}^{(0)}, \mathbf{h})}{\partial \boldsymbol{\theta}} \\ &\quad + \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^{(z)}) \frac{\partial E(\mathbf{v}^{(z)}, \mathbf{h})}{\partial \boldsymbol{\theta}} \end{aligned} \quad (8)$$

where  $\mathbf{v}^{(0)}$  and  $\mathbf{v}^{(z)}$  denote the original observed data and the sample generated by the Gibbs chain at step  $z$ .

### B. Deep Auto-Encoder

The DAE is designed for dimensionality reduction. As presented in Fig. 2, the node number of each layer in DAE decreases as the learning moves to higher layers [20]. Therefore, the dimensionality of the data reduces layer-wisely. The DAE is similar to the auto-associative neural networks in structure, but trained in a very different way. To train the DAE, every two adjacent layers are worked as an RBM. The hidden states of the previous RBM act as the visible states of the next RBM from the bottom to top. This can be interpreted as a “pretraining” process. After that, the RBMs are unfolded to act as the feed-forward neural network and fine-tuned by the back-propagation method.

## III. DEEP LEARNING FOR REDUCED-ORDER MODELING

Considering a spatiotemporal process  $Y$ , it has infinite-dimensional property in space and time domain. To model the spatiotemporal dynamic system, the snapshots  $Y_s$  of the process are measured by the sensors on spatial grid ( $N_s$ ) for discretization. Following the discretization into states, the governing PDEs become a system of ordinary differential equations (ODEs) in time domain as [30]

$$F(t, Y_s, Y'_s, \dots, Y_s^{(n-1)}) = Y_s^{(n)} \quad (9)$$

where  $F$  is a set of nonlinear functions  $\{f_i\}$ . The dimensionality ( $N_s$ ) of spatial grid increases as the process dynamics get complex. This will lead to the huge number of ODEs for the online application. Therefore, the dimensionality of  $Y_s$  should be reduced to low-dimensional representations  $y$  and then (9) can be rewritten in the reduced form, from which the prediction of  $\hat{y}$  can be derived

$$G(t, y, y', \dots, y^{(n-1)}) = y^{(n)}. \quad (10)$$

### A. Deep Auto-Encoder-Based Low-Dimensional Representations

To reduce the  $Y_s$  to  $y$ , there are many dimension reduction techniques. For linear case, the principle component analysis or POD is usually used. However, the POD method may fail to handle some kind of the DPS for reduced-order modeling. In this paper, the DAE is proposed for the dimensionality reduction. The proposed method is demonstrated in Fig. 3 which includes three phases. In phase I, the high-dimensional snapshots  $Y_s$  are reduced to low-dimensional representations  $y$  by the projector  $L^+ : \mathbb{R}^{N_s \times t} \mapsto \mathbb{R}^{n_s \times t}$

$$y = L^+(Y_s) \quad (11)$$

where  $n_s$  is the reduced dimensionality. In phase II, the low-dimensional representations  $y$  is modeled using (10) in a data-based approach

$$G(t, \hat{y}, \hat{y}', \dots, \hat{y}^{(n-1)}) = \hat{y}^{(n)} \quad (12)$$

where  $\hat{G}$  will be identified to approximate the real  $G$ . After obtaining the prediction  $\hat{y}$ , it can be reconstructed to obtain

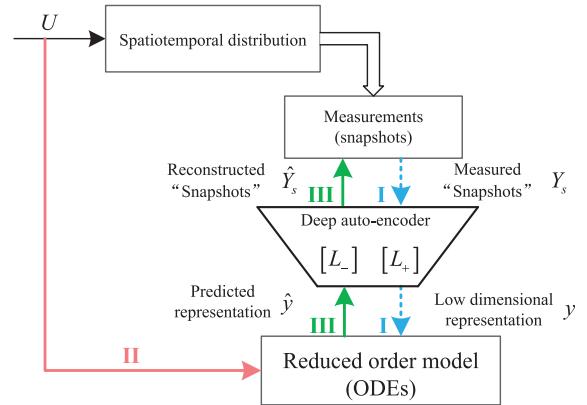


Fig. 3. Deep learning-based model reduction methodology.

predicted output  $\hat{Y}_s$  by reconstructor  $L^- : \mathbb{R}^{n_s \times t} \mapsto \mathbb{R}^{N_s \times t}$

$$\hat{Y}_s = L^-(\hat{y}). \quad (13)$$

The modeling error can be calculate as follows:

$$\varepsilon = \|Y_s - \hat{Y}_s\|_2 \quad (14)$$

where  $\|\cdot\|_2$  denotes the  $L_2$  norm. The minimization of  $\varepsilon$  is equivalent to find the optimal  $y$  which means the optimization of  $L^-$  and  $L^+$  should be conducted simultaneously. The DAE is thus proposed to identify the  $L^-$  and  $L^+$  so that they can be trained in an invertible way [20], [31] as follows:

$$\begin{aligned} L^+(Y) &= \sigma(W_k \sigma(W_{k-1} \cdots \sigma(W_1 Y + c_1) \cdots + c_{k-1}) + c_k) \\ L^-(y) &= \sigma(W_1^T \sigma(W_2^T \cdots \sigma(W_k^T y + b_k) \cdots + b_2) + b_1) \end{aligned} \quad (15)$$

where  $L^+$  and  $L^-$  uses the same set of weights  $W_i$  to ensure the simultaneous optimization,  $\sigma$  is the sigmoid function.

### B. Low-Dimensional Representations-Based Reduced-Order Modeling

After training the  $L^-$  and  $L^+$ , the low-dimensional representations  $y$  can be obtained. However, the representations  $y$  are a set of data which can not be used directly. The reduced order modeling should be carried out using (12). For the linear case, the identification of  $\hat{G}$  can be done with polynomial models of autoregressive integrated moving average structure as follows [32]:

$$A(q)y(t) = B(q)U(t) + \frac{C(q)}{1 - q^{-1}}e(t) \quad (16)$$

where  $q^{-1}$  is the backward shift operator and  $q^{-1}y(t) = y(t-1)$ ,  $U(t)$  is the system input,  $e(t)$  is the white noise disturbance and  $A(q)$ ,  $B(q)$ , and  $C(q)$  are defined as follows:

$$\begin{aligned} A(q) &= 1 + a_1q^{-1} + \cdots + a_nq^{-n_a} \\ B(q) &= b_1 + b_2q^{-1} + \cdots + b_{n_b}q^{-n_b+1} \\ C(q) &= 1 + c_1q^{-1} + \cdots + c_nq^{-n_c}. \end{aligned} \quad (17)$$

However, in many cases, the obtained ODEs are nonlinear. Thus, a more general nonlinear structure is developed as follows:

$$y(t) = A_w y_q + B_w U_q + H([y_q, U_q]) + C_w e \quad (18)$$

TABLE I  
IMPLEMENTATION OF THE PROPOSED METHOD

<b>Input:</b>	spatiotemporal $\hat{Y}_s$
<b>Output:</b>	parameter $[W_{level}, b_{level}, c_{level}]$ for $level = 1 : k$ and $\beta$
1	Normalize $\hat{Y}_s$ into $V$ and randomly divide it into $L$ mini-batches $S_1 \dots S_L$
2	Randomly initialize $[W_{level}, b_{level}, c_{level}]$ within $[0, 1]$ for $level = 1 : k$
3	Initialize $\Delta w_{ij} = \Delta b_i = \Delta c_j = 0$ for $i = 1 : n, j = 1 : m, level = 1$
4	<b>For</b> $l = 1 : L$ , <b>do</b>
5	<b>If</b> $level = 1$ , <b>do</b> $v^{(0)} \leftarrow S_l$ , <b>else do</b> $v^{(0)} \leftarrow [h_j, j = 1 : m]$ at $(level - 1)$
6	<b>For</b> $t = 1, \dots, z$ <b>do</b> ,
7	<b>For</b> $j = 1 : m$ , <b>do</b> sample $h_j^t \sim p(h_j   v^{(t)})$ by (5a)
8	<b>For</b> $i = 1 : n$ , <b>do</b> sample $v_i^{t+1} \sim p(v_i   h^{(t)})$ by (5b)
9	<b>For</b> $i = 1 : n, j = 1 : m$ , <b>do</b>
	$\Delta w_{ij} \leftarrow \Delta w_{ij} + p(h_j = 1   v^{(0)})v_i^{(0)} - p(h_j = 1   v^{(z)})v_i^{(z)}$ , $\Delta b_i \leftarrow \Delta b_i + v_i^{(0)} - v_i^{(z)}$ , $\Delta c_j \leftarrow \Delta c_j + p(h_j = 1   v^{(0)}) - p(h_j = 1   v^{(k)})$ by (8)
10	$W_{level} \leftarrow W_{level} + [\Delta w_{ij}]_{j=1:m}^{i=1:n}$
11	$b_{level} \leftarrow b_{level} + [\Delta b_i]_{i=1:n}$ , $c_{level} \leftarrow c_{level} + [\Delta c_j]_{j=1:m}$
12	<b>If</b> $level > limit$ , <b>then break out, else</b> $level = level + 1$ , <b>go to</b> 5
13	Fine-tuning the $[W_{level}, b_{level}, c_{level}]$ by back-propagation
14	$y \leftarrow \sigma(W_k \sigma(W_{k-1} \dots \sigma(W_1 V + c_1) \dots + c_{k-1}) + c_k)$
15	$x_q \leftarrow [y_q, U_q, \sigma(\varphi[y_q, U_q, e_q])]$
16	Calculate the $\beta \leftarrow [A_w, B_w, \Phi, C_w]$

where  $H$  is a nonlinear function that accounts the nonlinear relationship between  $y(t)$  and  $[y_q, U_q]$ . The state matrix is defined as follows:

$$\begin{aligned} y_q &= [y(t-1), \dots, y(t-(n_a+1))] \\ U_q &= [U(t), U(t-1), \dots, U(t-n_b)] \\ e_q &= [e(t), e(t-1), \dots, e(t-(n_c+1))] \end{aligned} \quad (19)$$

where  $n_a, n_b$ , and  $n_c$  accounts for the delay of  $y, U$ , and  $e$ , respectively. Then the parameter matrix can be defined as follows:

$$\begin{aligned} A_w &= [a_1, \dots, a_{n_a-1}, a_{n_a}]^T \\ B_w &= [b_1, \dots, b_{n_b-1}, b_{n_b}]^T \\ C_w &= [1, c_1, \dots, c_{n_c-1}, c_{n_c}]^T. \end{aligned} \quad (20)$$

In some cases, the system is autonomous and  $U(t)$  can be set to zero. An efficient method will be used for parameter identification of (18) in this paper. The nonlinear function  $H$  is designed as

$$H(x) = \Phi \sigma(\varphi x). \quad (21)$$

By sampling  $\varphi$  randomly from Gaussian noise, (18) can be trained as a random vector functional-link (RVFL) [33], [34]. The universal approximation capability and fast training speed of RVFL can be guaranteed [34], [35]. By substituting (19) into (21), it can be reformulated as

$$y = \beta x_q \quad (22)$$

where  $x_q$  and  $\beta$  are defined as follows:

$$\begin{aligned} x_q &= [y_q, U_q, \sigma(\varphi[y_q, U_q, e_q])] \\ \beta &= [A_w, B_w, \Phi, C_w]^T \end{aligned} \quad (23)$$

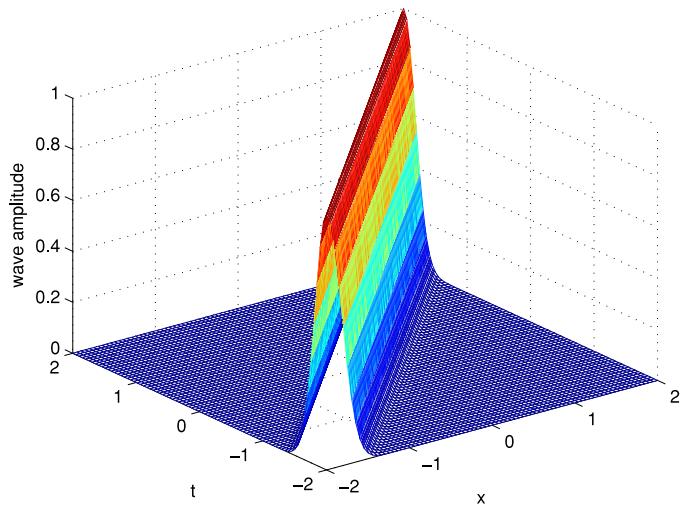


Fig. 4. Snapshots of  $Y_s$  derived by numerical method.

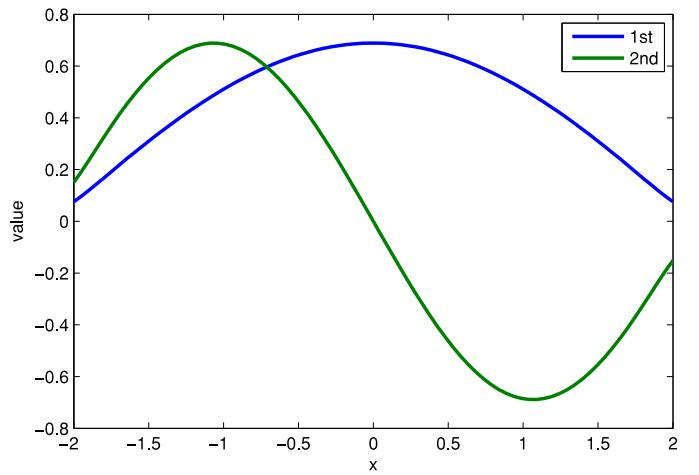


Fig. 5. Dominant spatial BFs identified by the POD method.

where  $\beta$  can be identified by least square method. In real application, online parameter identification should be carried out either by recursive least square [36] or sequential learning methods [37].

After the identification of  $\beta$ , the prediction of  $\hat{y}$  can be obtained to predict the spatiotemporal output  $\hat{Y}_s$  using (13). The implementation of the proposed method is illustrated in Table I.

#### IV. EXPERIMENTS

##### A. Advection Problem

The advection is a common transport mechanism of a substance by energies, such as heat, humidity, or salinity. The advection problem can be formulated as follows:

$$\frac{\partial Y}{\partial t} = -s \frac{\partial Y}{\partial x} \quad (24)$$

with the initial condition (IC):  $Y(x, 0) = f(x)$ , where  $s$  denote the transportation speed. This is a first-order hyperbolic PDE. The analytical solution is usually hard to be derived. The numerical solution is also not simple, even with one space

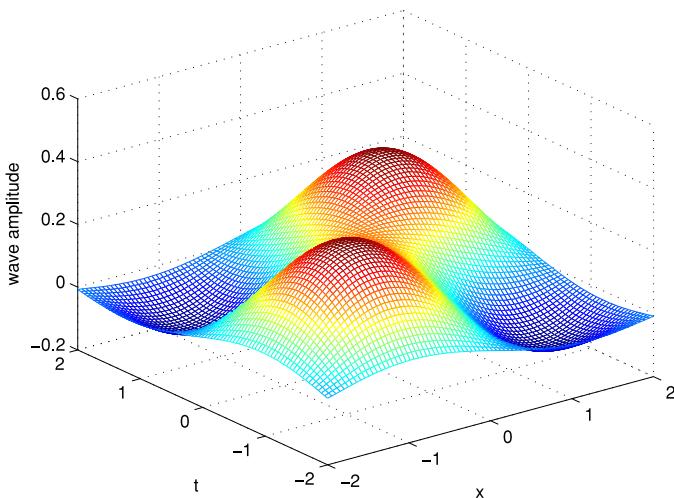
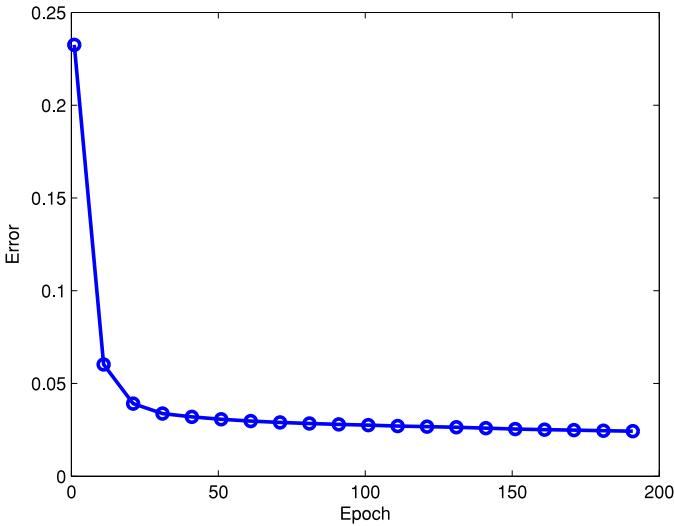
Fig. 6. Spatiotemporal prediction of  $Y_s$  by the POD method.

Fig. 7. RMSEs in epochs of the back propagation during the fine-tuning process.

dimension and a constant velocity field [38]. In this paper, a specific example with a smooth solution is considered by setting  $s$  to a constant and the IC function to Gaussian function

$$f(x) = e^{-\alpha x^2}. \quad (25)$$

Then the traveling wave solution of (23) can be obtained as [39]

$$Y(x, t) = e^{-\alpha(x-st)^2}. \quad (26)$$

The snapshot  $Y_s$  with a grid of 100 points along space domain  $x \in [-2, 2]$  and time domain  $t \in [-2, 2]$  are plotted in Fig. 4. Then the proposed model reduction method will compare with the POD method in terms of accuracy and efficiency.

The POD method is first applied to the snapshots  $Y_s$ . The spatial basis functions (BFs) of the first two dominant modes are shown in Fig. 5. With the spatial BFs, the space-time synthesis can be conducted to obtain the spatiotemporal distribution which is shown in Fig. 6. The POD method performs poorly in predicting the spatiotemporal dynamics  $Y_s$ .

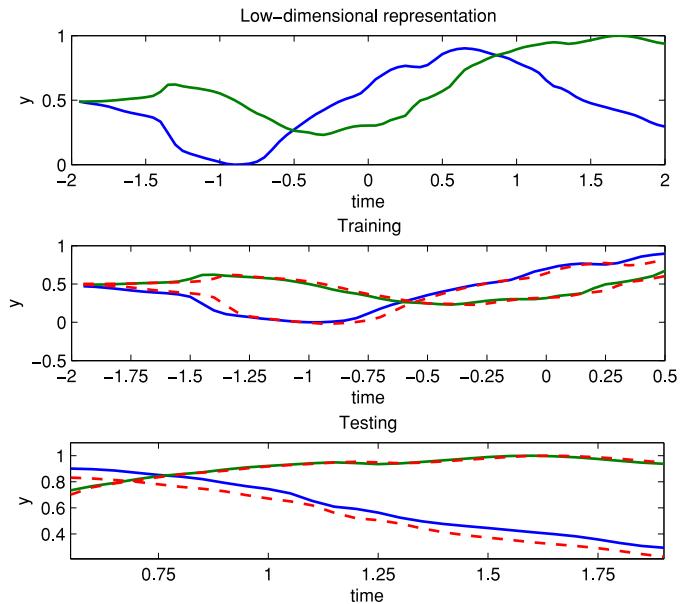
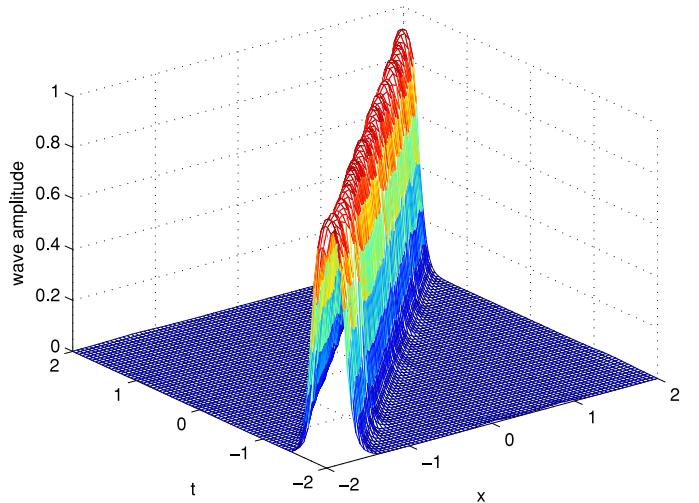


Fig. 8. Reduced-order modeling. Dashed lines denote the model prediction and the solid lines denote the real values.

Fig. 9. Spatiotemporal prediction of  $Y_s$  by the proposed method.

The proposed method is then applied. The training error during back-propagation is demonstrated in Fig. 7, which shows the convergence of the training process. The root mean squared errors are calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M (Y_s(x_i, t_j) - \hat{Y}_s(x_i, t_j))^2} \quad (27)$$

where  $M$  and  $N$  are the number of grids in space and time domain, respectively. After training process, the low dimensional representations  $y$  can be obtained as shown in Fig. 8. The data of  $y$  at time domain  $t \in [-2, 0.5]$  is used to train  $\beta$  in (22) and data at  $t \in [0.5, 2]$  for testing. The satisfactory training and testing results of the reduced-order model (10) are achieved and demonstrated in Fig. 8. Then the predicted  $\hat{y}$  is decoded to obtain  $\hat{Y}_s$  using  $L^-$  which is demonstrated in Fig. 9. It shows high reconstruction accuracy when compared to  $Y_s$ .

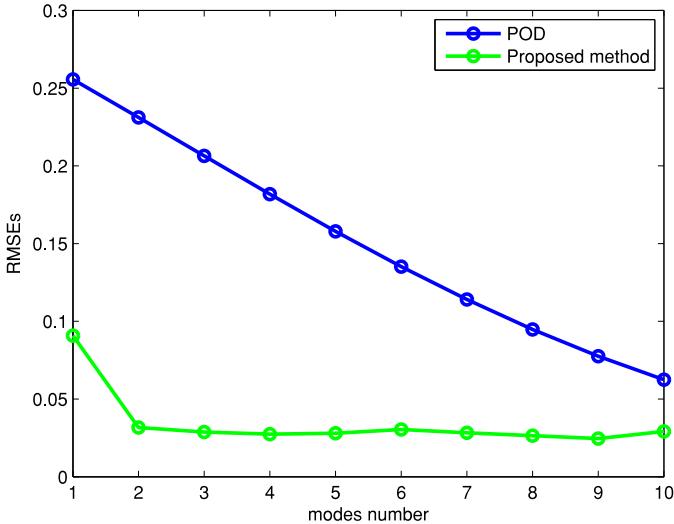


Fig. 10. Modeling accuracy comparisons between the POD method and the proposed method under different number of modes.

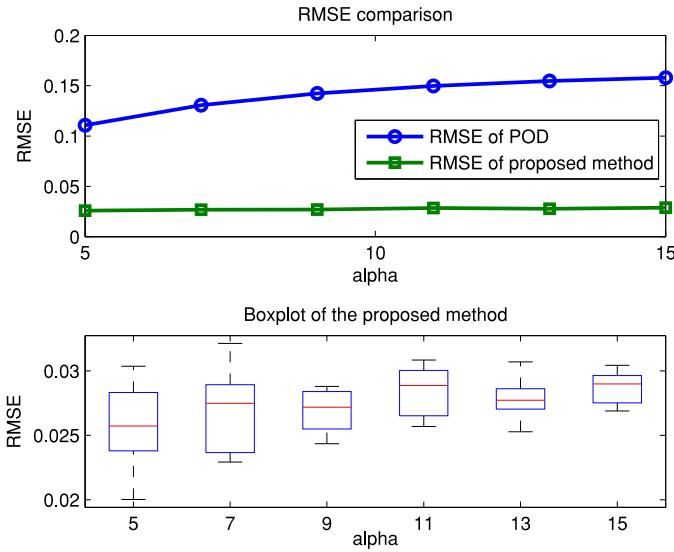


Fig. 11. Modeling accuracy comparison between the POD and the proposed method under different  $\alpha$  (top), boxplot for the modeling accuracy of the proposed method.

For the efficiency comparison of the two methods, it includes two aspects: 1) training cost and 2) modeling efficiency. In this case, the training time for POD and the proposed method are 0.02 s and 11.07 s, respectively. The training of POD method is only involves the single value decomposition, while the training of the proposed method concerns with gradients methods. Therefore, the train cost of the proposed method is obviously higher than the POD method but remains in an acceptable range.

The modeling efficiency is studied and presented in Fig. 10. The proposed method can achieve higher accuracy with fewer number of modes, while the POD performs poorly and increases slowly as the modes number increases. The proposed method demonstrated high model reduction efficiency in this paper.

The physical parameter  $\alpha$  plays an important role in the spatiotemporal dynamics. To show the robustness of the proposed

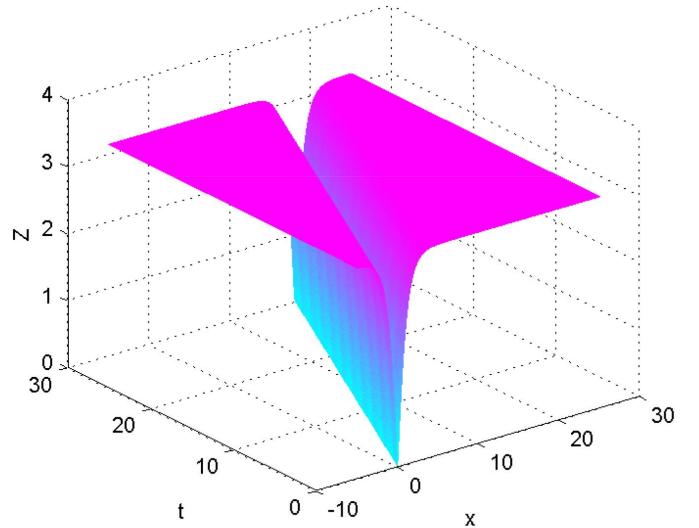


Fig. 12. Snapshots of  $Z_s$  derived by the numerical method.

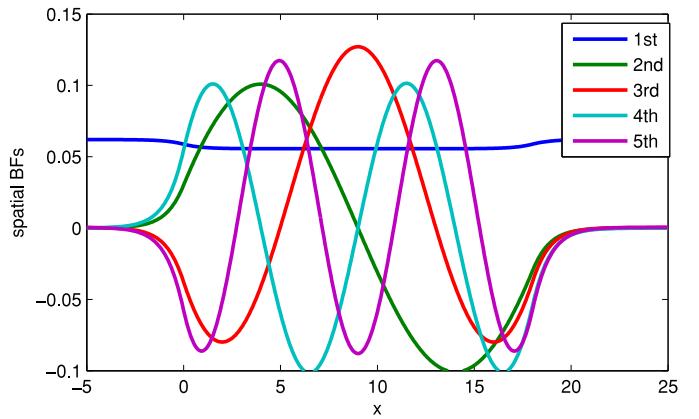


Fig. 13. First five spatial BFs identified by the POD method.

method,  $\alpha$  is set to increase from 5 to 15 with step size 2. Then the two methods are applied and the modes number are set to 5 for the fair comparison of their robustness. The RMSEs are calculated and presented in Fig. 11 which indicates that the accuracy of the POD method deteriorates as the  $\alpha$  increases, while the proposed is robust to the change of  $\alpha$ .

### B. Sine-Gordon Equation

The Sine-Gordon equation is a second-order hyperbolic PDE which commonly occurs in the research of fiber optics

$$\frac{\partial^2 Z}{\partial t^2} - \alpha^2 \frac{\partial^2 Z}{\partial x^2} = \beta \sin(Z). \quad (28)$$

The traveling wave solution of the PDE can be derived as [40]

$$Z = \arccos \left( 2 \tanh^2 \left( \frac{(-x + t) + \beta(x/\alpha)}{2k} \right) - 1 \right) \quad (29)$$

where  $\alpha$  is set to 1,  $\beta$  is set to 1 and  $k$  is set to 2 in this case. The snapshots of  $Z_s$  can be obtained as shown in Fig. 12. Then the snapshots are normalized to the range [0, 1] for the better comparison of the two method. In the experiment, the POD

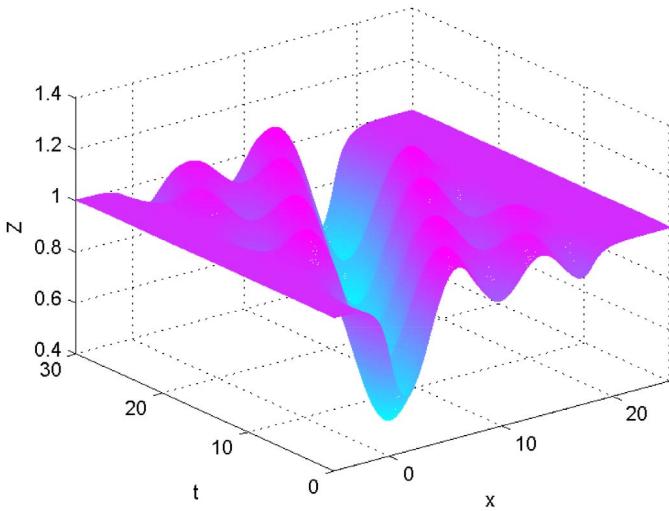
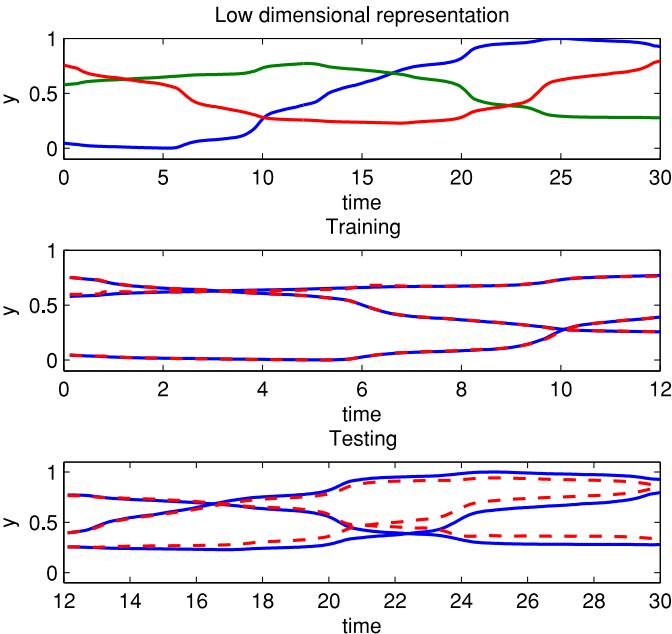
Fig. 14. Spatiotemporal prediction  $Z_s$  by the POD method.

Fig. 15. Reduced-order modeling by the proposed method. Dashed lines denote the model prediction and the solid lines denote the real values.

method needs 35 dominant spatial BFs (modes) to fully reconstruct the spatiotemporal dynamics. The first five dominant BFs are plot in Fig. 13. The spatiotemporal prediction with the first five BFs in the POD is shown in Fig. 14. The reduced order modeling is conducted in Fig. 15 with satisfactory results for both training and testing. The spatiotemporal prediction of the proposed method is presented in Fig. 16. By comparing Figs. 12, 14, and 16, the proposed method demonstrates better prediction accuracy than the POD method. The performances of the two method under different modes number are studied and compared in Fig. 17 which clearly demonstrates the higher modeling efficiency of the proposed method.

### C. Kolmogorov–Petrovskii–Piskunov Equation

In addition to the hyperbolic systems, the high dissipative systems are also studied. The POD method is suitable for

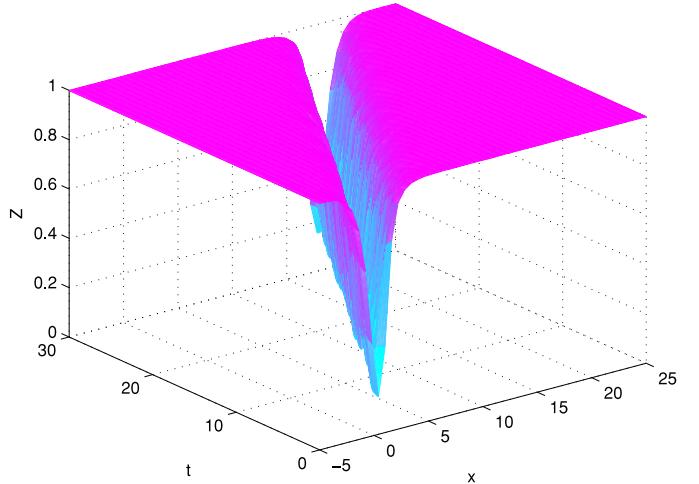
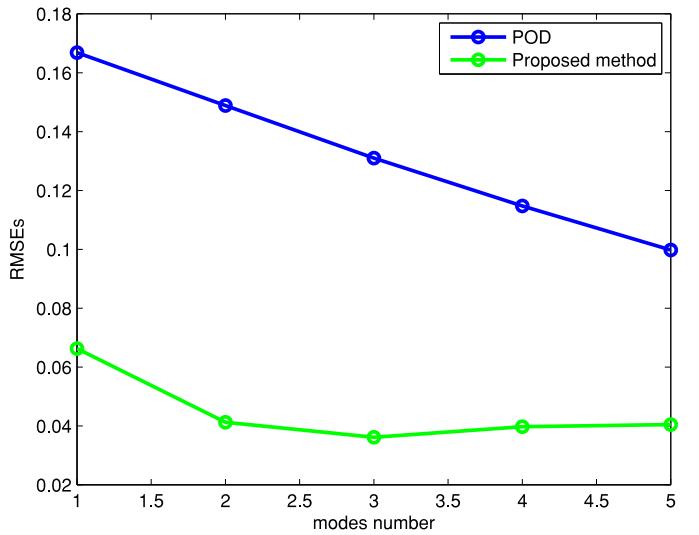
Fig. 16. Spatiotemporal prediction of  $Z_s$  by the proposed method.

Fig. 17. Modeling accuracy comparisons between the two methods under different modes number.

these systems since it can capture their dominant characteristic. Kolmogorov–Petrovskii–Piskunov equation is one of the high dissipative systems. It arises in the research of heat and mass transfer, combustion theory and many biology applications. It is a second-order parabolic PDE as follows [41]:

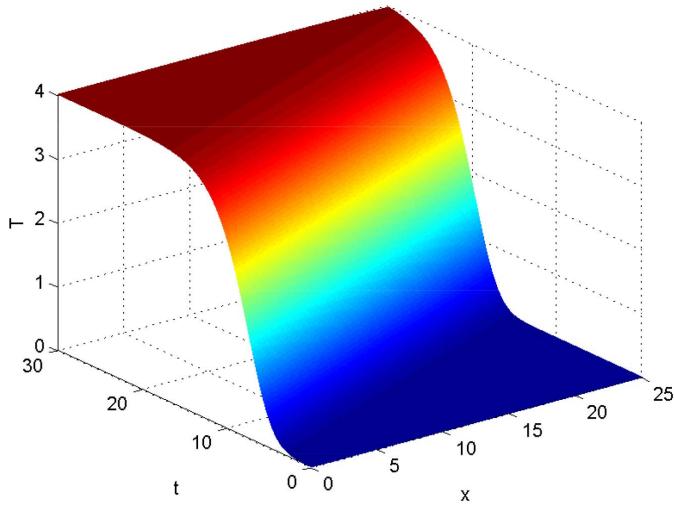
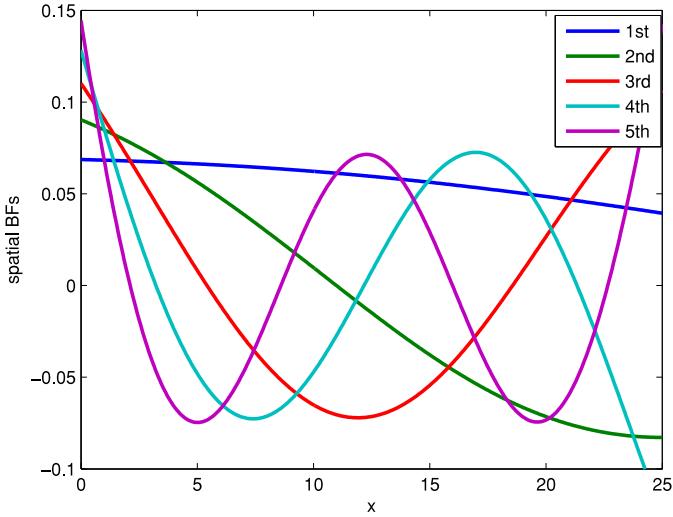
$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + aT + bT^m. \quad (30)$$

The traveling wave solution can be obtained as follows:

$$\begin{aligned} T(x, t) &= [\beta + Ce^{\lambda t \pm \mu x}]^{\frac{2}{1-m}} \\ T(x, t) &= [-\beta + Ce^{\lambda t \pm \mu x}]^{\frac{2}{1-m}} \end{aligned} \quad (31)$$

where  $C$  is an arbitrary constant,  $\lambda, \mu, \beta$  are given by

$$\begin{aligned} \lambda &= \frac{a(1-m)(m+3)}{2(m+1)} \\ \mu &= \sqrt{\frac{a(1-m)^2}{2(m+1)}} \\ \beta &= \sqrt{-\frac{b}{a}}. \end{aligned} \quad (32)$$

Fig. 18. Spatiotemporal snapshots of  $T_s$  derived by the numerical method.Fig. 19. First five dominant spatial BFs of  $T_s$  by the POD method.

In this paper,  $[a, b, m, C]$  are set to  $[1, -0.5, 1.5, 5]$ , respectively. The snapshots  $T_s$  can be obtained as Fig. 18. In this experiment, the dominant spatial BFs identified by the POD method are presented in Fig. 19. The spatiotemporal prediction by the POD method is presented in Fig. 20 which shows poor prediction of  $T_s$ . For the proposed method, the reduced-order model is identified as illustrated in Fig. 21. The corresponding spatiotemporal prediction is shown in Fig. 22 which demonstrates the high modeling accuracy of the proposed method. The performance under different modes number are studied in Fig. 23. The proposed method is able to capture most of the dynamics with fewer modes.

#### D. Kuramoto–Sivashinsky Equation

The Kuramoto–Sivashinsky (K–S) equation is also a high dissipative system and often used in extended physical systems such as phase dynamics in reaction-diffusion systems and fluctuations in fluid films [41]. 1-D K–S equation can be written

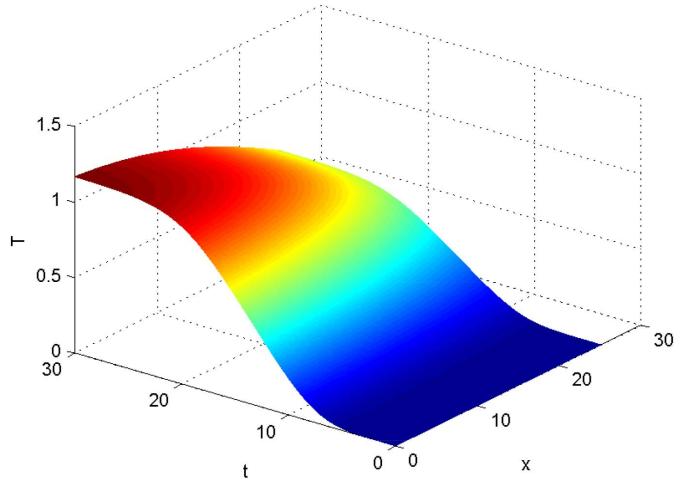
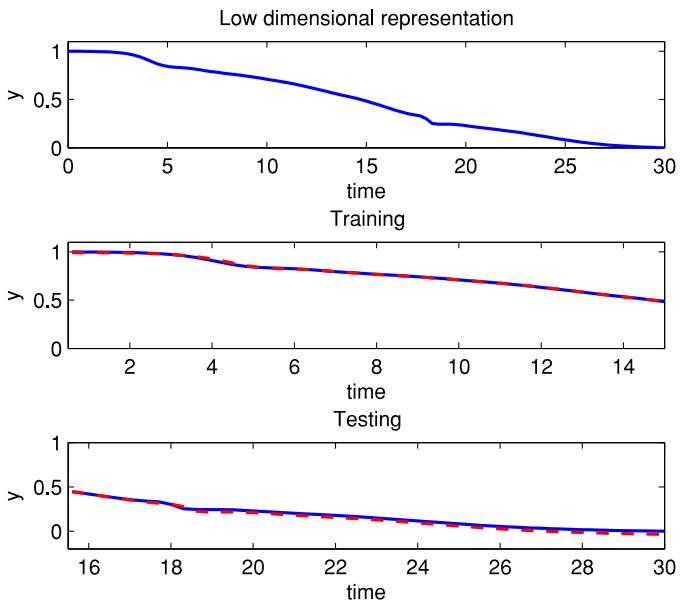
Fig. 20. Spatiotemporal prediction of  $T_s$  by the POD method.

Fig. 21. Reduced-order modeling by the proposed method. Dashed lines denote the model prediction and the solid lines denote the real values.

as follows:

$$\frac{\partial W}{\partial t} + W \frac{\partial W}{\partial x} + \frac{\partial^2 W}{\partial x^2} + \frac{\partial^4 W}{\partial x^4} = 0, x \in \left[ \frac{-L}{2}, \frac{L}{2} \right] \quad (33)$$

with boundary condition

$$W(x+L, t) = W(x, t) \quad (34)$$

where  $L$  is set to 22 in this paper. The snapshots  $W_s$  with a grid of 65 points along space domain  $x$  and time domain  $t \in [0, 500]$  are presented in Fig. 24. Then POD method and the proposed method are applied to model this complex spatiotemporal dynamics. The first five dominant spatial BFs are demonstrated in Fig. 25. With the spatial BFs, the prediction of  $W$  can be obtained.

For the proposed method, the low-dimensional representations can be obtained as Fig. 26. The first 200s data used

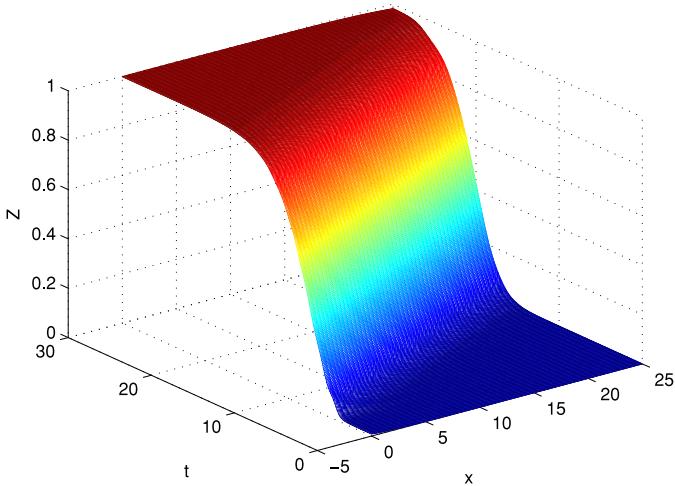


Fig. 22. Spatiotemporal prediction by the proposed method.

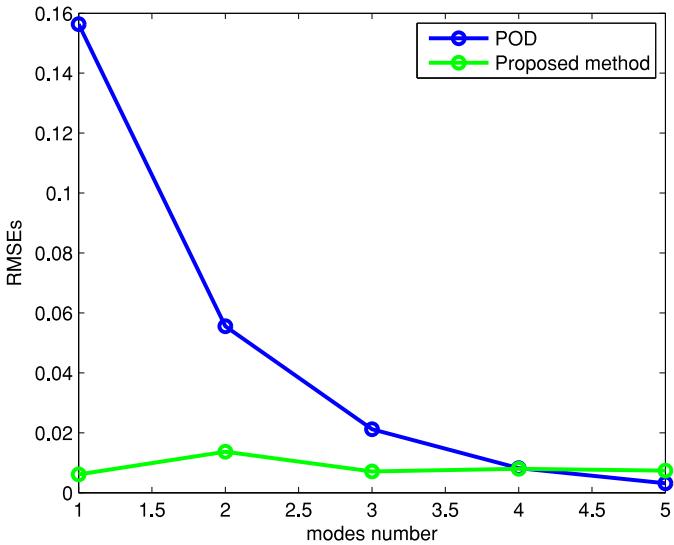
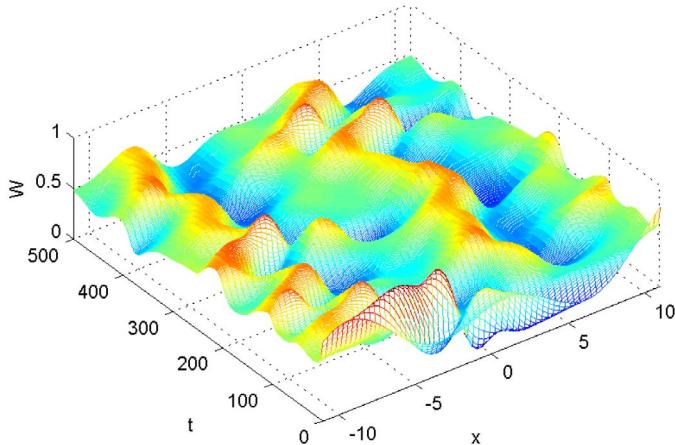


Fig. 23. Modeling accuracy comparisons between the two methods under different number of modes.

Fig. 24. Snapshots of  $W_s$  derived by numerical method.

as training and the following 300s data as testing. For a fair comparison, the number of modes is set to 2 for the two methods. The spatiotemporal prediction of the POD method and

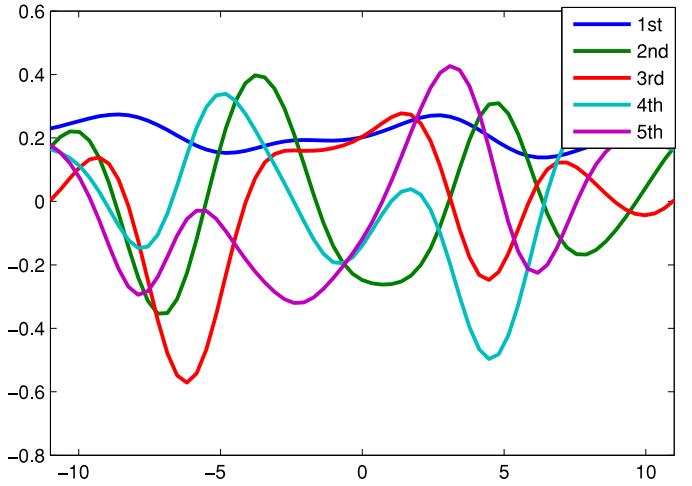
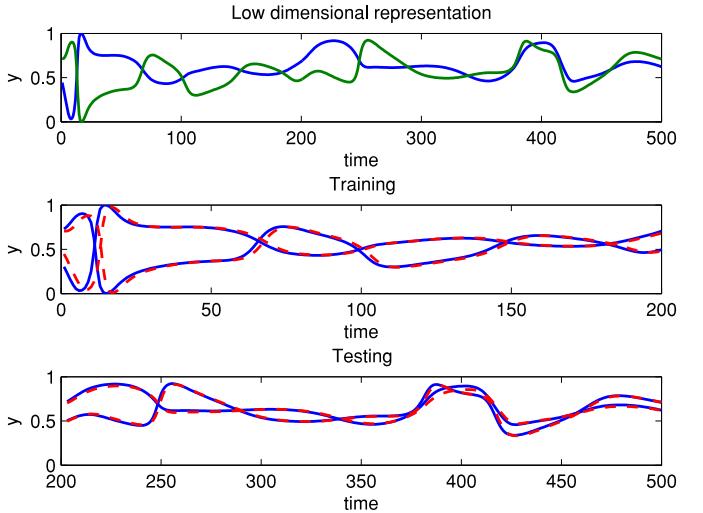
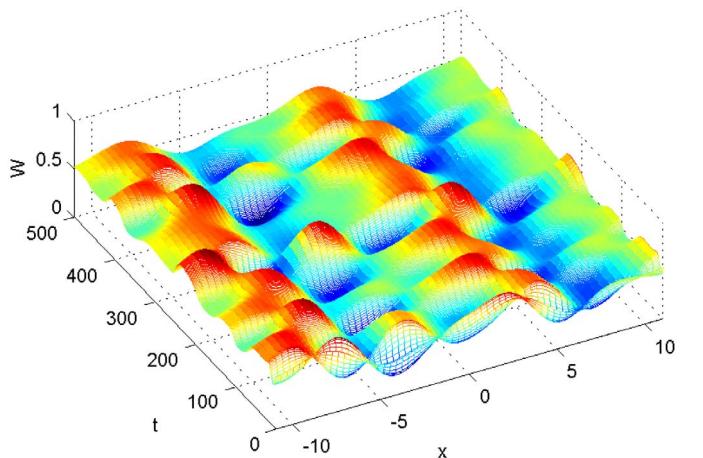
Fig. 25. First five dominant spatial BF of  $W_s$  by POD method.

Fig. 26. Reduced-order modeling by the proposed method. Solid lines denote the model prediction and the concrete lines denote the real values.

Fig. 27. Spatiotemporal prediction of  $W_s$  by the POD method.

the proposed method are presented in Figs. 27 and 28, respectively. The proposed method demonstrated better prediction results.

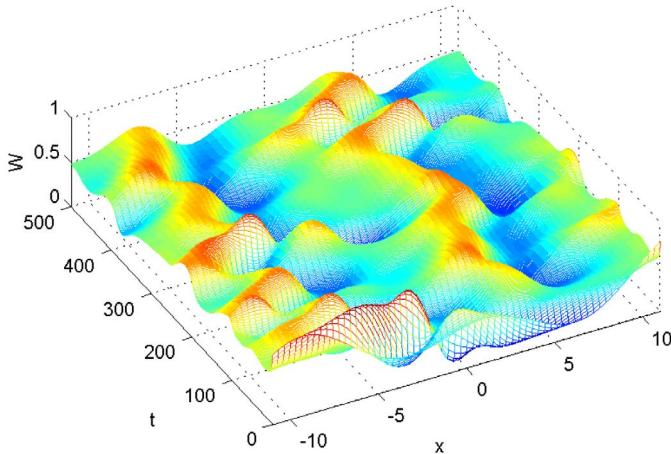


Fig. 28. Spatiotemporal prediction of  $W_s$  by the proposed method.

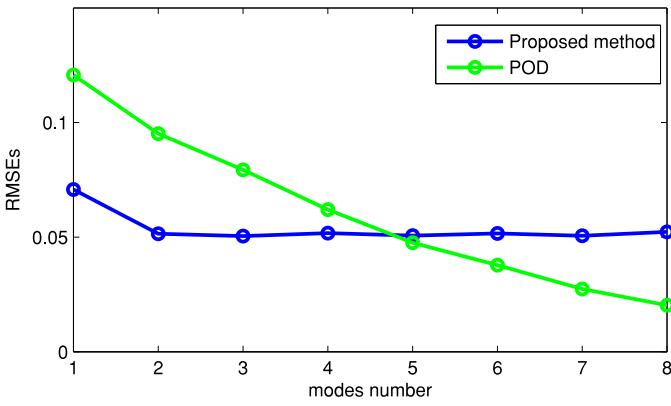


Fig. 29. Modeling accuracies with respect to modes number for the two methods.

The modeling accuracies under different mode numbers are studied and presented in Fig. 29. As indicated by Fig. 29, the POD method outperforms the proposed method when the nodes number exceeds 5.

## V. DISCUSSION

The proposed method shows prevailing performance when compared with the POD method in the experiments. For the hyperbolic systems, the POD method usually requires large number of modes to obtain satisfactory spatiotemporal prediction. Its performance also deteriorates as the physical parameter increases. While the proposed method is able to achieve high modeling accuracy with small number of modes and robust to the parameter changes. The POD is suitable for the high dissipative system, so it can efficiently capture the dominant dynamics of the system with relatively small number of modes. For these systems, the proposed method demonstrates better high modeling efficiency and modeling accuracy under limited number of modes or specific modeling accuracies. In general, the proposed method is very useful for the fast online applications where the high modeling efficiency and accuracy are required.

## VI. CONCLUSION

In this paper, a deep learning-based model reduction method has been proposed. A three-phase method is designed to obtain

the near-optimal solution for model reduction. Extensive experimental studies are conducted to demonstrate the effectiveness of the proposed method. Though the proposed method is relatively time consuming in training, it has better modeling accuracy and modeling efficiency than the POD method.

The proposed method is highly efficient in modeling the complex DPS, especially with nonparabolic features. It is suitable to the systems which require fast predictions of the spatiotemporal dynamics, for example it can be used to predict the electrochemical states distribution of the batteries in hybrid electrical vehicles. A promising further development of the proposed method is to construct new spatial BFs from the low-dimensional representation derived by the deep learning method.

## ACKNOWLEDGMENT

The authors would like to thank the valuable comments and help from anonymous reviewers and editors.

## REFERENCES

- [1] H.-Y. Zhu, H.-N. Wu, and J.-W. Wang, “ $H_\infty$  disturbance attenuation for nonlinear coupled parabolic PDE-ODE systems via fuzzy-model-based control approach,” *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.
- [2] A. R. Mitchell and D. F. Griffiths, *The Finite Difference Method in Partial Differential Equations*. New York, NY, USA: Wiley, 1980.
- [3] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, vol. 3. London, U.K.: McGraw-Hill, 1977.
- [4] M. Wang and H.-X. Li, “Spatiotemporal modeling of internal states distribution for lithium-ion battery,” *J. Power Sources*, vol. 301, pp. 261–270, Jan. 2016.
- [5] T. Xiao and H.-X. Li, “Sliding mode control design for a rapid thermal processing system,” *Chem. Eng. Sci.*, vol. 143, pp. 76–85, Apr. 2016.
- [6] W. H. Schilders, H. A. Van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13. Heidelberg, Germany: Springer, 2008.
- [7] J.-L. Wang, H.-N. Wu, T. Huang, S.-Y. Ren, and J. Wu, “Pinning control for synchronization of coupled reaction-diffusion neural networks with directed topologies,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 8, pp. 1109–1120, Aug. 2016.
- [8] S. A. Orszag, “Numerical methods for the simulation of turbulence,” *Phys. Fluids*, vol. 12, no. 12, p. II-250, 1969.
- [9] G. Berkooz, P. Holmes, and J. L. Lumley, “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annu. Rev. Fluid Mech.*, vol. 25, no. 1, pp. 539–575, 1993.
- [10] H.-X. Li and C. Qi, “Modeling of distributed parameter systems for applications—A synthesized review from time-space separation,” *J. Process Control*, vol. 20, no. 8, pp. 891–901, 2010.
- [11] M. D. Feit and J. A. Fleck, Jr., “Solution of the Schrödinger equation by a spectral method II: Vibrational energy levels of triatomic molecules,” *J. Chem. Phys.*, vol. 78, no. 1, pp. 301–308, 1983.
- [12] M. D. Feit, J. A. Fleck, Jr., and A. Steiger, “Solution of the Schrödinger equation by a spectral method,” *J. Comput. Phys.*, vol. 47, no. 3, pp. 412–433, 1982.
- [13] H.-N. Wu and S. Feng, “Guaranteed-cost finite-time fuzzy control for temperature-constrained nonlinear coupled heat-ODE systems,” *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.
- [14] P. Benner, S. Gugercin, and K. Willcox, “A survey of projection-based model reduction methods for parametric dynamical systems,” *SIAM Rev.*, vol. 57, no. 4, pp. 483–531, 2015.
- [15] A. Chatterjee, “An introduction to the proper orthogonal decomposition,” *Current Sci.*, vol. 78, no. 7, pp. 808–817, 2000.
- [16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [17] A. Fischer and C. Igel, “An introduction to restricted Boltzmann machines,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Heidelberg, Germany: Springer, 2012, pp. 14–36.

- [18] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Computer Vision–ECCV 2010*. Heidelberg, Germany: Springer, 2010, pp. 140–153.
- [19] J. Ngiam *et al.*, "Multimodal deep learning," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 689–696.
- [20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [21] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AICHE J.*, vol. 37, no. 2, pp. 233–243, 1991.
- [22] J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002.
- [23] Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski, "A theoretical framework for back-propagation," in *The Connectionist Models Summer School*, vol. 1. Pittsburg, CA, USA: Morgan Kaufmann, 1988, pp. 21–28.
- [24] D. Erhan *et al.*, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Feb. 2010.
- [25] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*. New York, NY, USA: Wiley, 1988.
- [26] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. Chichester, U.K.: Wiley, 2009.
- [27] W. R. Gilks, *Markov Chain Monte Carlo*. Hoboken, NJ, USA: Wiley, 2005.
- [28] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [29] C. K. Carter and R. Kohn, "On Gibbs sampling for state space models," *Biometrika*, vol. 81, no. 3, pp. 541–553, 1994.
- [30] A. LaBryer, P. J. Attar, and P. Vedula, "A framework for optimal temporal reduced order modeling of nonlinear dynamical systems," *J. Sound Vib.*, vol. 332, no. 4, pp. 993–1010, 2013.
- [31] C. Qi and H.-X. Li, "Nonlinear dimension reduction based neural modeling for distributed parameter processes," *Chem. Eng. Sci.*, vol. 64, no. 19, pp. 4164–4170, 2009.
- [32] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Stat. Assoc.*, vol. 65, no. 332, pp. 1509–1526, 1970.
- [33] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.
- [34] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.
- [35] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.
- [36] C. S. Leung, K. W. Wong, P. F. Sum, and L. W. Chan, "On-line training and pruning for recursive least square algorithms," *Electron. Lett.*, vol. 32, no. 23, pp. 2152–2153, Nov. 1996.
- [37] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.
- [38] J. W. Delleur, *The Handbook of Groundwater Engineering*. Boca Raton, FL, USA: CRC Press, 2006.
- [39] G. Griffiths and W. E. Schiesser, *Traveling Wave Analysis of Partial Differential Equations: Numerical and Analytical Methods With MATLAB and Maple*. Amsterdam, The Netherlands: Academic Press, 2010.
- [40] P. G. Drazin and R. S. Johnson, *Solitons: An Introduction*, vol. 2. Cambridge, U.K.: Cambridge Univ. Press, 1989.
- [41] A. D. Polianin and V. F. Zaitsev, *Handbook of Nonlinear Partial Differential Equations*. Boca Raton, FL, USA: CRC Press, 2004.



**Mingliang Wang** (S'16–M'16) received the B.E. degree from the China University of Mining and Technology, Xuzhou, China, in 2012, and the Ph.D. degree in systems engineering and engineering management from the City University of Hong Kong, Hong Kong, in 2016.

His current research interests include spatiotemporal modeling and online adaptive estimation for distributed parameter system, data-based modeling, and optimization.



**Han-Xiong Li** (F'11) received the B.E. degree in aerospace engineering from the National University of Defense Technology, Changsha, China, in 1982, the M.E. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1991, and the Ph.D. degree in electrical engineering from the University of Auckland, Auckland, New Zealand, in 1997.

He is currently a Professor with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong. He has a broad knowledge and experience in both academia and industry. He has authored two books, six patents, and published over 180 SCI journal papers with an H-index of 33. His current research interests include system intelligence and control, process design and control integration, and distributed parameter systems with applications to electronics packaging.

Dr. Li was a recipient of the Distinguished Young Scholar (overseas) by the China National Science Foundation in 2004, the Chang Jiang Professorship by the Ministry of Education, China in 2006, and the National Professorship in China Thousand Talents Program in 2010. He serves as an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS from 2009 to 2015. He serves as a Distinguished Expert for Hunan Government and China Federation of Returned Overseas.



**Xin Chen** received the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1995.

He is currently a Professor with the School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, China. His research interests include mechanical dynamics, IC packaging equipment technology, and precision machining and measurement.



**Yun Chen** received the B.S. and Ph.D. degrees in mechanical engineering from Central South University, Changsha, China, in 2009 and 2014, respectively.

He is currently a Lecturer with the School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, China. His research interests include modeling, microelectronics packaging technology, and equipment.