

A Sliding Window Based Dynamic Spatiotemporal Modeling for Distributed Parameter Systems With Time-Dependent Boundary Conditions

Bing-Chuan Wang¹ and Han-Xiong Li², *Fellow, IEEE*

Abstract—Time/space separation based spatiotemporal modeling methods have been proven to be effective and efficient for modeling a class of distributed parameter systems (DPSs). However, these conventional methods may not work satisfactorily for DPSs with time-dependent boundary conditions. A sliding window based dynamic spatiotemporal modeling method is proposed for this kind of DPSs. First, the sliding window is appropriately designed to capture the most recent spatiotemporal data. Then, the conventional Karhunen–Loève method can be used to construct the analytical model. Besides, a more general sliding window method can be achieved by using a forgetting factor to adjust different influence of the current and previous data. This analytical model can be utilized for online performance prediction. Simulation experiments on a benchmark and a battery with unknown boundary cooling have demonstrated the superior performance of the proposed method on the DPSs with time-dependent boundary conditions.

Index Terms—Distributed parameter systems (DPSs), forgetting factor, Karhunen–Loève (KL), sliding window, time-dependent boundary conditions.

I. INTRODUCTION

A PLENTY of physical and industrial processes, such as chemical reaction process [1], fluid flow [2], and thermal process [3] are distributed parameter systems (DPSs), which are governed by partial differential equations (PDEs) and exhibit spatiotemporal dynamics. Modeling DPSs is significant and

Manuscript received May 25, 2018; accepted July 15, 2018. Date of publication July 31, 2018; date of current version April 3, 2019. This work was supported in part by the General Research Fund Project from Research Grant Council of Hong Kong SAR (CityU: 11205615), and in part by the National Natural Science Foundations of China under Grant U1501248. Paper no. TII-18-1306. (*Corresponding author: Han-Xiong Li.*)

B.-C. Wang is with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong, and also with the State Key Laboratory of High Performance Complex Manufacturing, Central South University, Changsha 410083, China (e-mail: bingcwan3-c@my.cityu.edu.hk).

H.-X. Li is with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong (e-mail: mehqli@cityu.edu.hk).

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/TII.2018.2859444

essential for simulation, controlling, and optimization. However, it would be challenging to model the complicated spatiotemporal dynamic process due to the infinite-dimensional nature.

Time/space separation based modeling methods [4] have been proven to be effective and efficient to model DPSs with numerous results reported in the literature. When the DPS is known, a spectral method can be utilized to reduce it into a number of ordinary differential equations (ODEs). The spectral method is successfully integrated with neural network for modeling the complex curing process [3], where the spectral method is for time/space separation and the neural network for estimation of temporal nonlinearities. Different types of spatial activation functions are utilized for model reduction and extreme learning machine (ELM) [5] is for online estimating nonlinear ODEs [6]. When the DPS is unknown, the data-based time/space separation, such as the Karhunen–Loève (KL) method, is used to extract information from experimental data. Similar to the neural spectral method, KL can be integrated with the neural network to model nonlinear DPSs [7] with computational intelligence utilized to optimize the neural network. The famous ELM can also be combined with KL method for spatiotemporal modeling [8]. For the very complex DPSs, a different model configuration could be designed to match the dynamic complexity better. The spatial kernels using Volterra series [9], [10] and Hammerstein series [11] are designed to capture very complex spatiotemporal dynamics. For coupled nonlinear dynamics, the dual extended Kalman filter method is designed to model the internal states of the battery system [12]. When there are strong stochastic uncertainties, the least-squares support vector machine (LS-SVM) [13] and Gaussian process [14] could be utilized for temporal modeling. Also, there is a nonlinear time/space separation modeling method using locally linear embedding idea [15] to model the thermal process of lithium-ion batteries [16], and a dual LS-SVM is designed for the coupled dynamics of the curing process [17].

From the literature review, it can be found that most of the time/space separation based methods work for DPSs with time-invariant boundary conditions. However, DPSs with time-dependent boundary conditions, which exists widely in the industrial processes, are scarcely taken into account. When

modeling a DPS online, conventional time/space separation based methods would take all collected data equally to construct an analytic model. This manner is effective to DPSs with time-invariant boundary conditions due to the fact that all data are generated from the same PDE with the same boundary conditions. However, it would lose its effectiveness on DPSs with time-dependent boundary conditions. Because previous data cannot reflect the properties of the DPS with current boundary conditions. Given this, it is necessary to attempt to tailor time/space separation based method properly to model DPSs with time-dependent boundary conditions. Wang and Li [18] incorporate the variations of time-dependent boundary conditions into the reduced model. Without modifying the dominant spatial basis functions (SBFs), the performance of this method may be limited to a great extent. Armaou and Christofides [19] transform the time-dependent system into a time-invariant one by using an analytical expression that describes how the spatial domain changes. However, this information is not available in many real-world applications. Narasingam *et al.* [20] propose a temporal-clustering based method to address this issue, where the snapshots are classified into several clusters, and an analytical model is constructed for each cluster. Since the clustering process involves an optimization problem, its efficiency will be limited to some extent. To address the challenge introduced by the time-dependent boundary conditions, Izadi and Dubljevic [21] design a mapping function that maps the time-dependent domain to a fixed referenced one. To construct the mapping function, the time-dependent boundary conditions usually need to be known. Though a grid-point based method is introduced to calculate the mapping function numerically, meshing itself would be time-consuming. Recently, Sidhu *et al.* [22] combine the sparse proper orthogonal decomposition and Galerkin method to model a hydraulic fracturing process. Since Galerkin method is utilized, information of some parameters of the process must be known. As can be seen, most methods for DPSs with time-dependent boundary conditions require some parameters or analytical expression of the considered system to be known, which may not be available in most practical cases. Besides, some clustering and mapping methods would be time-consuming. Hence, designing a simple yet effective data-driven method is promising.

In recent years, the successful utilization of the sliding window method has been seen in various scientific fields, such as activity recognition [23], channel identification [24], and protein annotations extraction [25]. As a data-driven method, the sliding window method is easy to implement and efficient. Besides, by a sliding window, the time-varying properties can be elaborately caught. Although it has numerous advantages, the sliding window method has not been utilized for modeling of DPSs (to the best of our knowledge). By a sliding window, the most recent data, which truly reflects the real-time intrinsic nature of a process, will be utilized for modeling. Thus, it would be a good option for modeling DPSs with time-dependent boundary conditions.

Based on the above discussions, a sliding window based dynamic KL modeling method (SW-KL) is proposed for DPSs with time-dependent boundary conditions. A proper window is

designed to capture the most critical spatiotemporal dynamics with others ignored. Then, the conventional KL methods can be utilized to construct the analytical model. Because the window is moving over time, the recent and most important data relevant to the process will be captured, and the old data will be dropped to improve the effectiveness and computation efficiency. In order to bridge the gap between SW-KL and conventional KL method, a forgetting vector is introduced, by which the influence of all data can be appropriately adjusted over time. The proposed method is simple and ease of implementation for industrial application. A benchmark DPS with time-dependent boundary conditions is chosen to validate the effectiveness of the proposed SW-KL. In addition, SW-KL is also utilized to predict the thermal distribution of a lithium-ion battery under unknown boundary cooling. Experimental results demonstrate its overwhelming advantage over the conventional KL method. The main contributions of this paper are summarized as follows.

- 1) This paper makes the first attempt to utilize the sliding window technique for spatiotemporal modeling.
- 2) SW-KL is a data-driven method and does not require the analytical expression of a system, which implies that it is very proper for industrial applications.
- 3) Experiments on a benchmark and a lithium-ion battery have validated the effectiveness of SW-KL.

The rest of this paper is organized as follows. Section II is the problem description. The proposed SW-KL is elaborated in Section III. Section IV is the experimental study, and the concluding remarks are summarized in Section V.

II. PROBLEM DESCRIPTION

A typical DPS that exhibits spatiotemporal dynamics is formulated as follows:

$$\frac{\partial y(x, t)}{\partial t} = \alpha \frac{\partial}{\partial x} \left(k(y(x, t)) \frac{\partial y(x, t)}{\partial x} \right) + \beta \frac{\partial y(x, t)}{\partial x} + f(y(x, t)) + \omega b(x)u(t) \quad (1)$$

with boundary and initial conditions

$$y(l_1(t), t) = B_1(t), \quad y(l_2(t), t) = B_2(t), \quad y(x, 0) = I(x)$$

where $x \in [l_1, l_2]$ is the spatial variable, t is the temporal variable, $y(x, t)$ is the spatiotemporal state, $u(t)$ is the control value, $b(x)$ is the spatial distribution matrix of $u(t)$, $k(y(x, t))$, and $f(y(x, t))$ are functions of spatiotemporal state and α , β , and ω , are constants. Combined with the corresponding initial condition, i.e., $y(x, 0) = I(x)$, and boundary conditions, i.e., $y(l_i(t), t) = B_i(t)$, $i \in \{1, 2\}$, the analytical or numerical solutions of the DPS can be derived. In addition, a boundary condition, i.e., $y(l_i(t), t) = B_i(t)$, is called time-dependent if and only if $l_i(t)$ or $B_i(t)$ is time-dependent.

Due to space/time coupled dynamics, DPSs always exhibit infinite-dimensional nature and strong nonlinearities that make them hard to model. When the boundary conditions are time-dependent, the situation would be more worse. A sliding window based dynamic spatiotemporal modeling method (SW-KL) will be designed to handle the DPSs with time-dependent boundary conditions for online application.

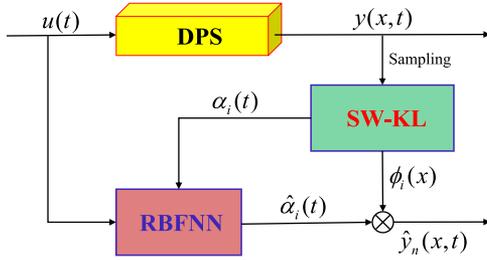


Fig. 1. Framework of the proposed SW-KL modeling method.

III. SLIDING WINDOW BASED DYNAMIC SPATIOTEMPORAL MODELING

A. Framework

The proposed SW-KL is a time/space separation method. As shown in Fig. 1, n dominant SBFs, i.e., $\phi_i(x)$, $i \in \{1, \dots, n\}$, and the corresponding temporal coefficients (time series), i.e., $\alpha_i(t)$, are extracted from the sampled spatiotemporal data. More specifically, the SW-KL method is utilized to obtain these dominant SBFs. Subsequently, the corresponding temporal coefficients can be achieved. Due to its super approximation ability, the radial basis function based neural network (RBFNN) [26], [27] is modified to construct the temporal model and then this temporal model will be employed to predict the future temporal coefficients, i.e., $\hat{\alpha}_i(t)$. Finally, by synthesizing the predicted temporal coefficients and dominant SBFs together, the predicted spatiotemporal states, i.e., $\hat{y}_n(x, t)$, can be attained.

The main elements of the proposed SW-KL will be elaborated in detail one by one. First, the sliding window method is presented. Afterwards, the forgetting factor, which is used to bridge the gap between SW-KL and conventional KL method, is introduced. Next, the RBFNN-based temporal model is elaborated. Finally, the heuristic algorithm, which is used to decide two parameters, i.e., the window size and the forgetting factor, is designed.

B. Sliding Window Method

The proposed method, i.e., SW-KL, is a time/space separation based modeling method. It shares the same framework with conventional KL modeling method (referred to Appendix A). The main difference between two methods is mainly lying on the data utilized for time/space separation.

As shown in Fig. 2, when modeling DPSs, conventional KL treats previously acquired data and newly coming data equally. It is very effective and accurate to DPSs with time-invariant boundary conditions. However, this manner might lose its effectiveness on modeling DPSs with time-dependent boundary conditions. When online modeling such DPSs, previously acquired data may be out of date. Therefore, fresh data should be carefully utilized. To this end, SW-KL designs a sliding window to obtain recent data for calculating dominant SBFs. By sliding this window along the time axis, the impact of previous data can be alleviated and then current local dynamics can be captured.

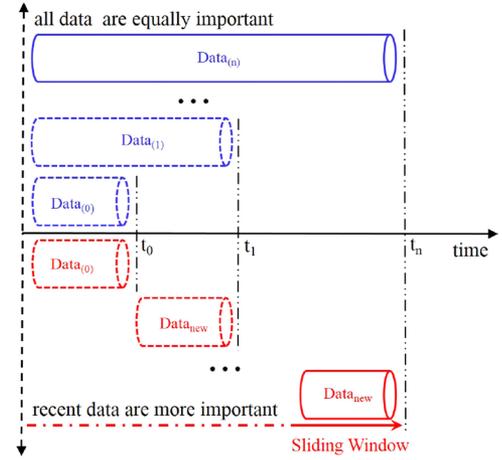


Fig. 2. Conceptual illustration of SW-KL.

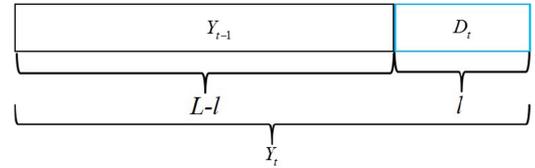


Fig. 3. Collected data at time t .

More specifically, as shown in Fig. 3, at time t , the conventional method utilizes all collected data, i.e., $Y_t \in \mathbb{R}^{N \times L}$, to calculate dominant SBFs and temporal coefficients. First, the spatial correlation matrix of Y_t is calculated:

$$\begin{aligned}
 \bar{R}_t &= \frac{1}{L} Y_t Y_t^T \\
 &= \frac{1}{L} [Y_{t-1} \ D_t] \begin{bmatrix} Y_{t-1}^T \\ D_t^T \end{bmatrix} = \frac{1}{L} (Y_{t-1} Y_{t-1}^T + D_t D_t^T) \\
 &= \frac{L-l}{L} \frac{1}{L-l} Y_{t-1} Y_{t-1}^T + \frac{l}{L} \frac{1}{l} D_t D_t^T \\
 &= \frac{L-l}{L} \bar{R}_{t-1} + \frac{l}{L} \hat{R}_t
 \end{aligned} \tag{2}$$

where Y_t is divided into two parts, i.e., previous data $Y_{t-1} \in \mathbb{R}^{N \times (L-l)}$ and newly collected data by the sliding window $D_t \in \mathbb{R}^{N \times l}$. Afterwards, the dominant SBFs and temporal coefficients can be attained according to (18) and (19) in Appendix A, respectively.

As discussed above, for DPSs with time-dependent boundary conditions, previous data may have a negative impact. In view of this, SW-KL discards the previous data, i.e., $Y_{t-1} \in \mathbb{R}^{N \times (L-l)}$, and utilizes the data in the sliding window, i.e., $D_t \in \mathbb{R}^{N \times l}$, to construct the current model. In this case, the spatial correlation matrix \hat{R}_t is utilized to calculate dominant SBFs and temporal coefficients.

C. Forgetting Factor

In order to bridge the gap between SW-KL and conventional KL, an adjustable forgetting factor is introduced. Specifically,

instead of discarding previous data totally, a forgetting factor μ ($0 \leq \mu \leq 1$) can be assigned to the previous spatial correlation matrix \bar{R}_{t-1} to alleviate its impact. So the spatial correlation matrix \bar{R}_t can be rewritten as

$$\bar{R}_t = \mu \frac{L-l}{L} \bar{R}_{t-1} + \frac{l}{L} \hat{R}_t. \quad (3)$$

Similarly, \bar{R}_{t-1} can be represented as

$$\bar{R}_{t-1} = \mu \frac{L-2l}{L-l} \bar{R}_{t-2} + \frac{l}{L-l} \hat{R}_{t-1}. \quad (4)$$

Furthermore, by combining (3) and (4), we can get

$$\bar{R}_t = \mu^2 \frac{L-2l}{L} \bar{R}_{t-2} + \mu \frac{l}{L} \hat{R}_{t-1} + \frac{l}{L} \hat{R}_t. \quad (5)$$

Expand this equation continually, and we can get

$$\bar{R}_t = \mu^t \frac{L-tl}{L} \bar{R}_0 + \frac{l}{L} \sum_{i=0}^{t-1} \mu^i \hat{R}_{t-i} \quad (6)$$

where \bar{R}_0 is the spatial correlation matrix of initial data and \hat{R}_{t-i} is the spatial correlation matrix of data in the sliding window at time $(t-i)$.

As can be seen, the weights of previous data, i.e., \bar{R}_0 and \hat{R}_{t-i} , decrease exponentially. In addition, it will be the case of sliding window method by setting $\mu = 0$ and the case of conventional KL modeling method by setting $\mu = 1$. As a result, the forgetting factor can bridge the gap between sliding window method and conventional KL modeling method. That is to say, by utilizing the forgetting factor, a more general sliding window method is achieved.

D. RBFNN-Based Temporal Model Construction

As shown in Fig. 1, after extracting $\phi_i(x)$ and $\alpha_i(t)$, $i \in \{1, \dots, n\}$ from selected data, the RBFNN [26], [27] is utilized to construct the temporal model. Mathematically, RBFNN uses a weighted sum of simple basis functions to interpolate the scatter points. Given the data points $\{(\vec{x}_i, \vec{y}_i) | \vec{x}_i \in \mathbb{R}^D, \vec{y}_i \in \mathbb{R}^d, i = 1, \dots, M\}$, RBFNN maps the inputs to the outputs as follows:

$$Y = \begin{bmatrix} K(\vec{x}_1 - \vec{c}_1) & \cdots & K(\vec{x}_1 - \vec{c}_m) \\ \vdots & \ddots & \vdots \\ K(\vec{x}_M - \vec{c}_1) & \cdots & K(\vec{x}_M - \vec{c}_m) \end{bmatrix} \times \begin{bmatrix} w_{1,1} & \cdots & w_{d,1} \\ \vdots & \ddots & \vdots \\ w_{1,m} & \cdots & w_{d,m} \end{bmatrix} \quad (7)$$

where $Y = [\vec{y}_1, \dots, \vec{y}_M]^T$, $K(\cdot)$ is the radial basis function (RBF), \vec{c}_h and $w_{j,h}$, $j \in \{1, \dots, d\}$, $h \in \{1, \dots, m\}$ are the center of the h th hidden neuron and the weight between the h th hidden neuron and the j th output neuron, respectively. When training data are available, the weights, i.e., $w_{j,h}$, can be approximated. Comprehensive details of RBFNN can be referred to [26], and the universal approximation ability of RBFNN is demonstrated in [27].

Algorithm 1: SW-KL.

Input: spatial correlation matrix at time $t-1$: \bar{R}_{t-1} ;

newly collected data at time t :

$D_t = \{y(x_i, t_j)\}_{i=1, j=L-l+1}^{N, L}$;

forgetting factor: μ

time lags: n_a, n_b ;

input control signal: $u(t)$;

Output: predicted spatiotemporal states;

1 Calculating spatial correlation matrix:

$\bar{R}_t = \mu \frac{L-l}{L} \bar{R}_{t-1} + \frac{l}{L} D_t D_t^T$;

2 Eigenvalue decomposition: $\bar{R}_t \bar{\phi}_i = \lambda_i \bar{\phi}_i$, $i = 1, \dots, K$
/* K is the rank of \bar{R}_t */

3 Sorting spatial basis functions $\bar{\phi}_i$ in a descending order according to their eigenvalues λ_i , $i = 1, \dots, K$;

4 Selecting the first n sorted spatial basis functions as dominant spatial basis functions, $\bar{\phi}_i$, $i = 1, \dots, n$;

5 Calculating temporal coefficients: $\bar{\alpha}_i = \bar{\phi}_i^T D_t$,
 $i = 1, \dots, n$;

6 RBFNN based temporal model construction;

7 Temporal predicting;

8 Spatiotemporal synthesis: $y(x_k, t_j) \approx \sum_i^n \phi_i(x_k) \alpha_i(t_j)$;

In order to model the temporal coefficients of a DPS by RBFNN, the following points should be carefully featured.

- 1) The input vector is decided as $\vec{x}_i = [\alpha_1(t-1), \dots, \alpha_1(t-na), \dots, \alpha_n(t-1), \dots, \alpha_n(t-na), u_1(t-1), \dots, u_1(t-nb), \dots, u_s(t-1), \dots, u_s(t-nb)]^T$, where na and nb denote the maximum input and control lags, respectively, n is the number of dominant SBFs, s is the number of control values. It should be noted that both na and nb are set as 1 based on the correlation analysis.
- 2) The output vector is the prediction at time t , which is set as $\vec{y}_i = \{\alpha_1(t), \dots, \alpha_n(t)\}$.
- 3) Note that the number of hidden neurons (i.e., m) will have a significant impact on the approximation accuracy. In this paper, m is set to 20 manually.
- 4) Gaussian RBF is adopted. The famous K -means clustering method [28] is utilized to seek the centers, i.e., \vec{c}_i , $i \in \{1, \dots, m\}$. Besides, the global width is selected as the average of all the Euclidian distances between the i th RBF center and its nearest neighbor.
- 5) Due to its efficiency, LASSO [29] is utilized to identify the weight matrix in (7).

Then the constructed RBFNN will be utilized for prediction. Finally, by spatiotemporal synthesis according to (14), the spatiotemporal states can be predicted.

In summary, the whole process of SW-KL modeling method, which utilizes SW-KL to obtain dominant SBFs and temporal coefficients and utilizes RBFNN to construct the temporal model, is summarized in Algorithm 1.

Remark: In this paper, our primary contribution is to propose a simple yet effective method to remedy the weakness of conventional modeling methods on DPSs with time-dependent boundary conditions. Although our main concentration is not put on theory, a concise proof of the boundedness of SW-KL is given in Appendix B.

E. Selection of Forgetting Factor and Window Size

As shown in (6), the spatial correlation matrix at time t , \bar{R}_t , can be affected by the magnitude of μ , the initial spatial correlation matrix \bar{R}_0 , and the spatial correlation matrix of data in the sliding window \hat{R}_{t-i} , $i = 0, \dots, t-1$. If t is large enough, the impact of \bar{R}_0 can be neglected due to that $0 \leq \mu \leq 1$. However, the significant impact of μ and \hat{R}_{t-i} cannot be eliminated. More specifically, the impact of these two components can be summarized as follows:

- 1) The smaller the μ is, the more quickly the impact of previous data decreases.
- 2) \hat{R}_{t-i} has a direct relation to window size l and the smaller the l is, the more quickly the impact of previous data decreases.

Thus, the forgetting factor μ and the window size l can adjust the impact of previous data that has a significant impact on the estimation accuracy. It is worth to note that the impact of previous data is varied on different industrial processes. So the selection of u and l is not trivial. In addition, the interaction between u and l would make this problem more intractable.

The selection of μ and l is an optimization problem in nature. Given a pair of (μ, l) , a candidate model can be constructed. And its performance can be evaluated as follows:

$$f(\mu, l) = \sqrt{\frac{1}{nl} \sum_{i=1}^n \sum_{j=1}^l (y(x_i, t_j) - \hat{y}(x_i, t_j))^2} \quad (8)$$

where $\{y(x_i, t_j)\}_{i=1, j=1}^{n, l}$ is a validation set sampled from the considered process and $\hat{y}(x_i, t_j)$ is the value approximated by the candidate model. By a proper optimization method, a satisfied pair of (μ, l) can be achieved.

Evolutionary algorithms (EAs) [30] have gained immense reputation in tackling complex optimization problems in recent years. Although the optimum cannot be guaranteed theoretically, EAs can always offer a satisfied local optimum with limited computation resources. Among all kinds of EAs, differential evolution (DE) [31] has shown its outperformed performance on continuous optimization problems. So, a DE variant is designed here to select a proper pair of (μ, l) :

Step 1) Initialization:

Step 1.1) Generate a random population with NP solutions, $P = \{(\mu_1, l_1), \dots, (\mu_{NP}, l_{NP})\}$ and evaluate the population according to (8), $FV = \{f(\mu_1, l_1), \dots, f(\mu_{NP}, l_{NP})\}$.

Step 1.2) Set $t = 1$, where t is the generation number.

Step 2) Updating:

For $i = 1, \dots, NP$ do

Step 2.1) If $\text{rand}(0, 1) < \frac{1}{1 + e^{-20(t/T - 0.5)}}$, utilize "DE/current-to-best/1/bin" to generate an offspring $(\hat{\mu}_i, \hat{l}_i)$, otherwise, utilize "DE/current-to-rand/1" to generate the offspring, where T is the predefined total number of generations.

Step 2.2) Evaluate the offspring according to (8), $f(\hat{\mu}_i, \hat{l}_i)$.

Step 2.3) If $f(\hat{\mu}_i, \hat{l}_i) < f(\mu_i, l_i)$, replace (μ_i, l_i) with $(\hat{\mu}_i, \hat{l}_i)$ and replace $f(\mu_i, l_i)$ with $f(\hat{\mu}_i, \hat{l}_i)$.

Step 3) Stopping criteria: If $t > T$, stop the evolution process and output the local optimum configuration of (μ^*, l^*) , otherwise, set $t = t + 1$ and go to step 2).

The computation complexity of this DE variant is analyzed as follows. In each generation, the DE operator would be

executed NP times. In addition, the whole algorithm would last T generations. Hence, the total complexity of this algorithm is $O(T \cdot NP)$. It should be noted that T is the total number of generations that need to be set manually. By setting it properly, a satisfied local optimum can be achieved.

IV. EXPERIMENTAL STUDY

In the experimental section, a benchmark DPS with time-dependent/time-invariant boundary conditions is chosen to evaluate the performance of SW-KL. Afterwards, it is compared with those of the conventional KL modeling method and the domain transformation method (abbreviated as DT) [1].¹ Besides, to further demonstrate its advantages, SW-KL is also employed to predict the thermal distribution of a lithium-ion battery under unknown boundary cooling. For all experiments, the root mean squared error (RMSE) is utilized as the performance index:

$$\text{RMSE} = \sqrt{\frac{1}{NL} \sum_{i=1}^N \sum_{j=1}^L (y(x_i, t_j) - \hat{y}(x_i, t_j))^2} \quad (9)$$

where $y(x_i, t_j)$ is the true spatiotemporal state at space point x_i and time t_j . $\hat{y}(x_i, t_j)$ is the corresponding estimated spatiotemporal state.

A. Experiments on the Benchmark DPS With Time-Dependent Boundary Conditions

The diffusion-reaction process with time-dependent boundary conditions [1] is selected to validate the performance of SW-KL:

$$\begin{aligned} \frac{\partial y(x, t)}{\partial t} &= \frac{\partial}{\partial x} \left(k(x) \frac{\partial y(x, t)}{\partial x} \right) + \beta_T(x) (e^{\frac{-r}{1+y(x, t)}} - e^{-r}) \\ &+ \beta_u(b(x)u(t) - y(x, t)) \end{aligned} \quad (10)$$

with initial and boundary conditions

$$y(x, 0) = 0.5, \quad y(0, t) = 0, \quad y(l(t), t) = 0.$$

Corresponding parameters are set as

$$l(t) = \pi(1.4 - 0.4e^{-0.02t^{2.7}}), \quad \beta_u = 2, \quad r = 4$$

$$\beta_T(x) = 45(1.5 - e^{-0.5x}), \quad k(x) = e^{-0.5x}$$

$$u(t) = [u_1(t), \dots, u_4(t)], \quad b(x) = [b_1(x), \dots, b_4(x)]$$

where $u_i(t) = 1.1 + (4 + 2 * \text{rand})e^{\frac{-i}{5}} \sin(500t/7 + 2 * \text{rand}) - 0.4e^{\frac{-i}{20}} \sin(500t + 2 * \text{rand})$ and $b_i(x) = H[x - (i-1)\pi/4] - H[x - i\pi/4]$ with $H(\cdot)$ being the standard Heaviside function. As shown in Fig. 4, time-dependent boundary conditions result that the spatial domains of the diffusion-reaction process are time-varying. Hence, this property makes it difficult to model.

All related parameters are set as follows. A total of 18 sensors are distributed uniformly along the space. A noise-free dataset of 1200 snapshots is collected from the system according to (10). The dimensionless sampling time ∇t is 0.01 s and simulation period is 12 s. White noise bounded by $0.02 \times (\max(y) - \min(y))/3$

¹We implement this method by using domain transformation to obtain dominant SBFs and utilizing RBFNN to construct the temporal model.

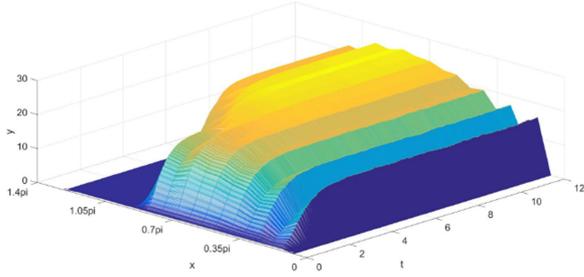


Fig. 4. Diffusion-reaction process with time-dependent boundary conditions.

TABLE I

PERFORMANCE COMPARISON ON THE DIFFUSION-REACTION PROCESS BASED ON AVERAGE RMSE OVER 2–12 s

method	conventional KL	DT	SW-KL
RMSE	7.805	2.6893	0.8086

TABLE II

PERFORMANCE COMPARISON ON THE DIFFUSION-REACTION PROCESS BASED ON AVERAGE RMSE OVER 10–12 s

method	conventional KL	DT	SW-KL
RMSE	4.285	0.4666	0.1312

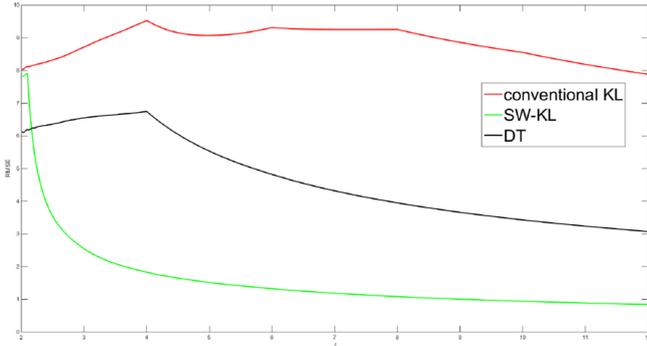


Fig. 5. Modeling RMSE from 2 to 12 s.

with 0 mean is added to the noise-free data to obtain the noisy output. The first 200 noisy snapshots are utilized to construct the nominal model while the rest 1000 noisy snapshots are employed for online spatiotemporal predicting. The window size l is set as 10. The forgetting factor μ is set as 0.2. The number of dominant SBFs is set as 2. Each experiment is repeated 50 independent runs.

The spatiotemporal modeling RMSEs of three methods are summarized in Tables I and II and Fig. 5, respectively. Obviously, SW-KL outperforms conventional KL modeling method on the diffusion-reaction process with time-dependent boundary conditions. Though the DT method outperforms SW-KL at the beginning, it will lose effectiveness to some extent when time goes on. Besides, when the DT method is implemented, the boundary conditions need to be known. It will limit DT's application in practical cases to a great extent.

Furthermore, the modeling error from 2 to 12 s and the modeling error from 10 to 12 s are described in Figs. 6 and 7,

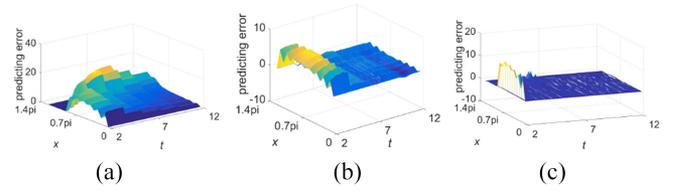


Fig. 6. Modeling error from 2 to 12 s. (a) Conventional KL. (b) DT. (c) SW-KL.

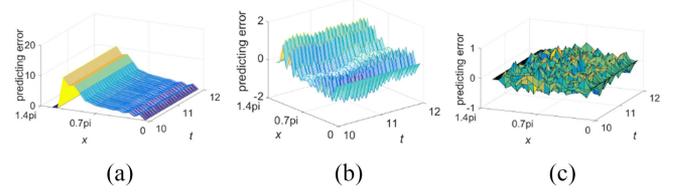


Fig. 7. Modeling error from 10 to 12 s. (a) Conventional KL. (b) DT. (c) SW-KL.

respectively. As shown in the figures, conventional KL modeling method performs extremely bad on time-dependent spatial domains while the proposed SW-KL remedy its weakness to a great extent. The DT method can outperform conventional KL modeling method. However, it is still worse than SW-KL.

Additionally, to test SW-KL's performance on DPS with time-invariant boundary conditions, we set $l(t) = 1.4\pi$. The same $u(t)$ is utilized to excite the DPS described as (10), and some data are sampled for modeling. Note that, the forgetting factor μ is set to 1 to match with the process. The prediction RMSEs of the conventional KL, the DT method, and SW-KL are 0.0552, 0.0583, and 0.0563, respectively. Hence, through setting the forgetting factor properly, SW-KL can model DPSs with time-invariant boundary conditions successfully.

To investigate the effectiveness of LASSO, we implement SW-KL by replacing LASSO with group LASSO. Afterwards, it is evaluated on the benchmark with time-dependent boundary conditions. The RMSEs from 2–12 s and 10–12 s are 0.8172 and 0.1293, respectively. The experimental result shows that no significant improvement presents. The reason may be that the basic LASSO can catch the main characteristic of the temporal model adequately.

B. Prediction of the Thermal Distribution of a Lithium-Ion Battery Under Unknown Boundary Cooling

The thermal distribution of a lithium-ion battery is governed by a PDE system [8]:

$$\rho C_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial T}{\partial y} \right) + q$$

$$q = aJ \left(\phi_{oc} - \phi - T \frac{d\phi_{oc}}{dT} \right) + a_p r_p i_p^2 + a_n r_n i_n^2$$
(11)

with boundary condition

$$-\lambda_x \frac{\partial T}{\partial x} - \lambda_y \frac{\partial T}{\partial y} = h(T - T_{air})$$
(12)

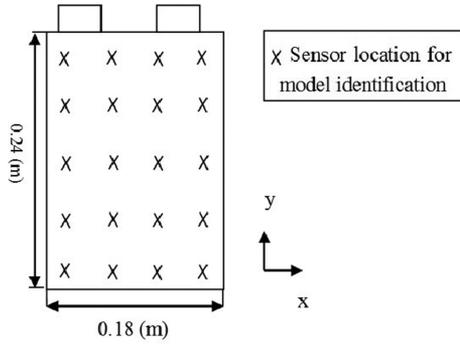


Fig. 8. Schematic diagram of the battery.

TABLE III
PREDICTION RMSE OF THE DISTRIBUTION OF BATTERY TEMPERATURE
OVER 200–850 S

method	conventional KL	SW-KL
RMSE	0.3556	0.0673

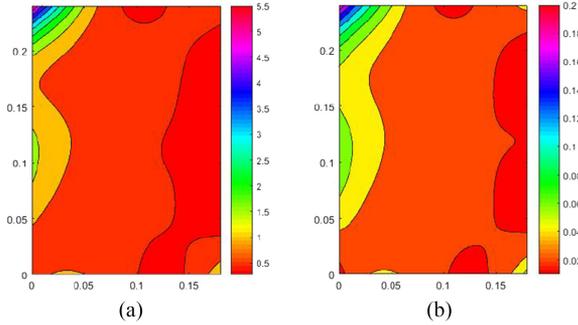


Fig. 9. Prediction error at 450 s. (a) Conventional KL. (b) SW-KL.

where T is the time/space coupled temperature, ρ is the battery density, C_p is the specific heat capacity, q is the heat generation rate, T_{air} is the environment temperature. In this paper, the convection coefficient h is varied due to unknown boundary cooling. The values of corresponding parameters are given as follows: $\rho = 10^3 \text{ kg/m}^3$, $C_p = 10^3 \text{ J/(kg} \cdot \text{K)}$, $T_{\text{air}} = 23.5^\circ\text{C}$, $a = 10/\text{m}$, $a_p = 26.6/\text{m}$, $a_n = 10.2/\text{m}$, $r_p = 9 \text{ E-5}$, $r_n = 1.207 \text{ E-4}$, and $k_x = k_y = \lambda_x = \lambda_y = 80 \text{ W/(m} \cdot \text{K)}$.

A lithium-ion battery, whose material of the negative electrode is Li_xC₆ and the material of the positive electrode is Li_xMn₂O₄, is used in this paper. Its schematic diagram is described in Fig. 8. In this experiment, a current of 3C rate is utilized. A total of 20 sensors are placed uniformly on the surface of the battery for data sampling. A data matrix with 5 rows and 4 columns is sampled every 1 s, and the sampling process lasts for 850 s. In addition, the data of first 200 s are utilized to construct the nominal model. The optimized μ and l are 0.3 and 10, respectively.

The experimental results are summarized in Table III. Besides, the distributed prediction errors of two methods at 450, 650, and 850 s are described in Figs. 9–11, respectively. Experimental results demonstrate the effectiveness and efficiency of SW-KL.

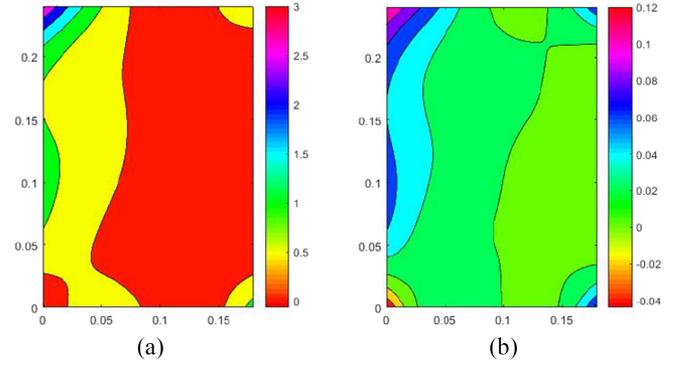


Fig. 10. Prediction error at 650 s. (a) Conventional KL. (b) SW-KL.

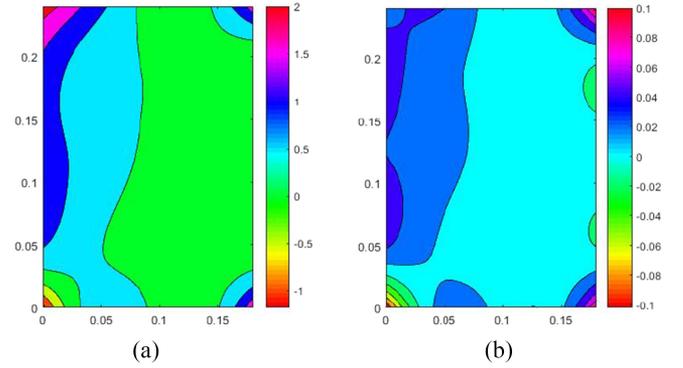


Fig. 11. Prediction error at 850 s. (a) Conventional KL. (b) SW-KL.

V. CONCLUSION

A SW-KL method is proposed for DPSs with time-dependent boundary conditions in this paper. A sliding window is designed to capture the most important information from newly coming data and ignore the old data that is outside of the window. The conventional KL method can be applied inside the window to construct a feasible analytical process model with high efficiency. A forgetting factor is also used to produce a more general window where the influence of all data can be adjusted exponentially over time. An efficient differential evolution variant is designed to select the forgetting factor and the window size. Experimental comparisons demonstrate the superior performance of the proposed method and its potential in industrial applications. In the future work, utilizing SW-KL for control problem would be taken into consideration. Besides, the incremental method will be utilized to further improve the efficiency of SW-KL.

APPENDIX A KL MODELING METHOD

For simplicity, define the inner product, norm, and ensemble average as: $(f(x), g(x)) = \int_{\Omega} f(x)g(x)d_x$, $\|f(x)\| = \sqrt{(f(x), f(x))}$ and $\langle f(x, t) \rangle = \frac{1}{L} \sum_{t=1}^L f(x, t)$.

The spatiotemporal state $y(x, t)$ of a DPS is expanded by a series of SBFs $\{\phi_i(x)\}_i^\infty$ as follows:

$$y(x, t) = \sum_i^\infty \phi_i(x) \alpha_i(t) \quad (13)$$

where $\alpha_i(t)$ is the temporal coefficient of $\phi_i(x)$. Then, the system is truncated and represented by n dominant SBFs

$$y_n(x, t) = \sum_i^n \phi_i(x) \alpha_i(t). \quad (14)$$

Conventionally, the truncation process can be formulated as

$$\begin{aligned} & \min_{\phi_i(x)} \langle \|y(x, t) - y_n(x, t)\|^2 \rangle \\ \text{s.t. } & (\phi_i(x), \phi_j(x)) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad i, j = 1, \dots, n \end{aligned} \quad (15)$$

The Lagrangian augment for optimizing (15) is

$$L = \langle \|y(x, t) - y_n(x, t)\|^2 \rangle + \sum_{i=1}^n \lambda_i ((\phi_i(x), \phi_i(x)) - 1) \quad (16)$$

the necessary condition to obtain the optimal solutions is

$$\int_{\Omega} R(x, \zeta) \phi_i(\zeta) d\zeta = \lambda_i \phi_i(x) \quad (17)$$

where $R(x, \zeta) = \langle y(x, t), y(\zeta, t) \rangle$ is the correlation function of two spatial points, i.e., x and ζ . $\phi_i(x)$ is just the right i th eigenfunction of $R(x, \zeta)$ and λ_i is the eigenvalue.

In practice, KL decomposition modeling method is implemented in a discrete way. N sensors are utilized to collect data and L snapshots $Y = \{y(x_i, t_j)\}_{i=1, j=1}^{N, L}$ are sampled. By discretizing (17), we can derive that

$$\bar{R} \bar{\phi}_i = \lambda_i \bar{\phi}_i \quad (18)$$

where $\bar{R} = \frac{1}{L} Y Y^T$ is the spatial correlation matrix and $\bar{\phi}_i$ is the i th discrete SBF. Eigenvalue decomposition [32] of \bar{R} or SVD decomposition [33] of Y can be utilized for calculating the SBFs $\{\bar{\phi}_i\}_{i=1}^K$, where K is the rank of \bar{R} .

For practical utilization, a small number $n < K$ of dominant SBFs are selected usually. We can arrange the eigenvalues $\lambda_1 > \lambda_2 > \dots > \lambda_K$ and eigenfunctions $\bar{\phi}_1(x), \bar{\phi}_2(x), \dots, \bar{\phi}_K(x)$ in order of the magnitude of the eigenvalues. Then, the first n SBFs are selected to capture 99% of the system energy.

Due to that dominant SBFs are orthogonal to each other, corresponding temporal coefficients can be derived as follows:

$$\alpha_i(t) = (\phi_i(x), y_n(x, t)). \quad (19)$$

And discrete realization is described as follows:

$$\bar{\alpha}_i = \bar{\phi}_i^T Y. \quad (20)$$

After the truncation, the infinite-dimensional DPS is reduced to n temporal ODE models $\alpha_i(t)$, $i \in \{1, 2, \dots, n\}$. Related parameters of the ODEs can be estimated from the low-dimensional input/output dataset by using temporal

predicting methods, such as least squares [34], neural networks [35], and fuzzy models [36]. Then, the obtained ODE model can be employed for prediction. Finally, the predicted spatiotemporal dynamics can be obtained by spatiotemporal synthesis.

APPENDIX B ERROR BOUND OF SW-KL

In practice, KL decomposition is implemented by principal component analysis. In the case, we will prove that SW-KL is bounded as follows.

Theorem 1: The reconstruction error of SW-KL is bounded, i.e., $\frac{1}{l} \sum_{i=1}^l \|d_i - \Phi \bar{\alpha}_i\|_2^2 \leq \epsilon$, where $D_t = [d_1, \dots, d_l]$, $\Phi = [\bar{\phi}_1, \dots, \bar{\phi}_n]$, and ϵ is a finite positive value.

Proof: In SW-KL, $(\frac{\mu}{L} Y_{t-1} Y_{t-1}^T + \frac{1}{L} D_t D_t^T)$ is utilized to calculate Φ . We can rearrange this formula as follows:

$$\begin{aligned} \frac{\mu}{L} Y_{t-1} Y_{t-1}^T + \frac{1}{L} D_t D_t^T &= \frac{1}{L} (\sqrt{\mu} Y_{t-1}) (\sqrt{\mu} Y_{t-1})^T + \frac{1}{L} D_t D_t^T \\ &= \frac{1}{L} \bar{Y}_{t-1} \bar{Y}_{t-1}^T + \frac{1}{L} D_t D_t^T. \end{aligned} \quad (21)$$

Hence, the Φ calculated by (21) is also the dominant SBFs of $[\bar{Y}_{t-1} D_t]$, where $\bar{Y}_{t-1} = \sqrt{\mu} Y_{t-1}$. This procedure is exactly the conventional principal component analysis [37]. Hence, the reconstruction error of $[\bar{Y}_{t-1} D_t]$ is bounded, which can be described as follows:

$$\frac{1}{L} \sum_{i=1}^{L-l} \|\bar{y}_i - \Phi \bar{\alpha}_i\|_2^2 + \frac{1}{L} \sum_{i=1}^l \|d_i - \Phi \bar{\alpha}_i\|_2^2 \leq \epsilon$$

where $\bar{Y}_t = [\bar{y}_1, \dots, \bar{y}_{L-l}]$ and ϵ is a finite positive value. By setting $\epsilon = \frac{l}{L} (\epsilon - \frac{1}{L} \sum_{i=1}^{L-l} \|\bar{y}_i - \Phi \bar{\alpha}_i\|_2^2)$, we can get that

$$\frac{1}{l} \sum_{i=1}^l \|d_i - \Phi \bar{\alpha}_i\|_2^2 \leq \epsilon. \quad \blacksquare$$

Theorem 2: The approximation error of SW-KL is bounded, i.e., $\frac{1}{l} \sum_{i=1}^l \|d_i - \Phi \hat{\alpha}_i\|_2^2 \leq \delta$, where $\hat{\alpha}_i$ is the temporal coefficient vector approximated by RBFNN and δ is a finite positive value.

Proof: The approximation error is calculated as follows:

$$\begin{aligned} \frac{1}{l} \sum_{i=1}^l \|d_i - \Phi \hat{\alpha}_i\|_2^2 &\leq \frac{1}{l} \left(\sum_{i=1}^l \|d_i - \Phi \bar{\alpha}_i + \Phi \bar{\alpha}_i - \Phi \hat{\alpha}_i\|_2 \right)^2 \\ &\leq \frac{1}{l} \left(\sum_{i=1}^l \|d_i - \Phi \bar{\alpha}_i\|_2 + \|\Phi \bar{\alpha}_i - \Phi \hat{\alpha}_i\|_2 \right)^2. \end{aligned}$$

Because $\frac{1}{l} \sum_{i=1}^l \|d_i - \Phi \bar{\alpha}_i\|_2^2 \leq \epsilon$, $\|d_i - \Phi \bar{\alpha}_i\|_2 \leq \sqrt{l\epsilon}$. That is to say, $\|d_i - \Phi \bar{\alpha}_i\|_2 \leq \sqrt{l\epsilon}$.

Consequently, we can obtain that

$$\begin{aligned} \frac{1}{l} \sum_{i=1}^l \|d_i - \Phi \hat{\alpha}_i\|_2^2 &\leq \frac{1}{l} \left(l\sqrt{l\epsilon} + \sum_{i=1}^l \|\Phi \bar{\alpha}_i - \Phi \hat{\alpha}_i\|_2 \right)^2 \\ &= \frac{1}{l} \left(l\sqrt{l\epsilon} + \sum_{i=1}^l \sqrt{(\bar{\alpha}_i - \hat{\alpha}_i)^T \Phi^T \Phi (\bar{\alpha}_i - \hat{\alpha}_i)} \right)^2 \\ &= \frac{1}{l} \left(l\sqrt{l\epsilon} + \sum_{i=1}^l \|\bar{\alpha}_i - \hat{\alpha}_i\|_2 \right)^2. \end{aligned}$$

The RBFNN can approximate any functions to an arbitrary accuracy [26], [27], [38]. That is to say, $\|\bar{\alpha}_i - \hat{\alpha}_i\|_2 \leq \rho$, where ρ is a finite positive value. Consequently, we can obtain that

$$\begin{aligned} \frac{1}{l} \sum_{i=1}^l \|d_i - \Phi \hat{\alpha}_i\|_2^2 &\leq \frac{1}{l} \left(l\sqrt{l\epsilon} + \sum_{i=1}^l \|\bar{\alpha}_i - \hat{\alpha}_i\|_2 \right)^2 \\ &\leq l(\sqrt{l\epsilon} + \rho)^2. \end{aligned}$$

By setting $\delta = l(\sqrt{l\epsilon} + \rho)^2$, we can derive that

$$\frac{1}{l} \sum_{i=1}^l \|d_i - \Phi \hat{\alpha}_i\|_2^2 \leq \delta. \quad \blacksquare$$

REFERENCES

- [1] A. Armaou and P. D. Christofides, "Finite-dimensional control of nonlinear parabolic PDE systems with time-dependent spatial domains using empirical eigenfunctions," *Int. J. Appl. Math. Comput. Sci.*, vol. 11, no. 2, pp. 287–318, 2001.
- [2] J. Baker, A. Armaou, and P. D. Christofides, "Nonlinear control of incompressible fluid flow: application to burgers' equation and 2d channel flow," *J. Math. Anal. Appl.*, vol. 252, no. 1, pp. 230–255, 2000.
- [3] H. Deng, H.-X. Li, and G. Chen, "Spectral-approximation-based intelligent modeling for distributed thermal processes," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 5, pp. 686–700, Sep. 2005.
- [4] 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.
- [5] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.
- [6] X. Lu, F. Yin, C. Liu, and M. Huang, "Online spatiotemporal extreme learning machine for complex time-varying distributed parameter systems," *IEEE Trans. Ind. Inform.*, vol. 13, no. 4, pp. 1753–1762, Aug. 2017.
- [7] R. Zhang, J. Tao, R. Lu, and Q. Jin, "Decoupled ARX and RBF neural network modeling using PCA and GA optimization for nonlinear distributed parameter systems," *IEEE Trans. Neural Netw. Learn.*, vol. 29, no. 2, pp. 457–469, Feb. 2018.
- [8] Z. Liu and H.-X. Li, "Extreme learning machine based spatiotemporal modeling of lithium-ion battery thermal dynamics," *J. Power Sources*, vol. 277, pp. 228–238, 2015.
- [9] H.-X. Li, C. Qi, and Y. Yu, "A spatio-temporal volterra modeling approach for a class of distributed industrial processes," *J. Process Control*, vol. 19, no. 7, pp. 1126–1142, 2009.
- [10] C. Cheng, X. Dong, Z. Peng, W. Zhang, and G. Meng, "Wavelet basis expansion-based spatio-temporal volterra kernels identification for nonlinear distributed parameter systems," *Nonlinear Dyn.*, vol. 78, no. 2, pp. 1179–1192, 2014.
- [11] C. Qi and H.-X. Li, "A time/space separation-based Hammerstein modeling approach for nonlinear distributed parameter processes," *Comput. Chem. Eng.*, vol. 33, no. 7, pp. 1247–1260, 2009.
- [12] M. Wang and H.-X. Li, "Spatiotemporal modeling of internal states distribution for lithium-ion battery," *J. Power Sources*, vol. 301, pp. 261–270, 2016.
- [13] C. Qi, H.-X. Li, X. Zhang, X. Zhao, S. Li, and F. Gao, "Time/space-separation-based SVM modeling for nonlinear distributed parameter processes," *Ind. Eng. Chem. Res.*, vol. 50, no. 1, pp. 332–341, 2010.
- [14] P. Sun, J. Chen, and L. Xie, "Self-active and recursively selective Gaussian process models for nonlinear distributed parameter systems," *Chem. Eng. Sci.*, vol. 123, pp. 125–136, 2015.
- [15] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [16] Z. Liu and H.-X. Li, "A spatiotemporal estimation method for temperature distribution in lithium-ion batteries," *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2300–2307, Nov. 2014.
- [17] K.-K. Xu, H.-X. Li, and H.-D. Yang, "Dual least squares support vector machines based spatiotemporal modeling for nonlinear distributed thermal processes," *J. Process Control*, vol. 54, pp. 81–89, 2017.
- [18] M. Wang and H.-X. Li, "Real-time estimation of temperature distribution for cylindrical lithium-ion batteries under boundary cooling," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2316–2324, Mar. 2017.
- [19] A. Armaou and P. D. Christofides, "Computation of empirical eigenfunctions and order reduction for nonlinear parabolic PDE systems with time-dependent spatial domains," *Nonlinear Anal.-Theor.*, vol. 47, no. 4, pp. 2869–2874, 2001.
- [20] A. Narasingam, P. Siddhamshetty, and J. Sang-II Kwon, "Temporal clustering for order reduction of nonlinear parabolic PDE systems with time-dependent spatial domains: Application to a hydraulic fracturing process," *AIChE J.*, vol. 63, no. 9, pp. 3818–3831, 2017.
- [21] M. Izadi and S. Dujljevic, "Order-reduction of parabolic PDEs with time-varying domain using empirical eigenfunctions," *AIChE J.*, vol. 59, no. 11, pp. 4142–4150, 2013.
- [22] H. S. Sidhu, A. Narasingam, P. Siddhamshetty, and J. S.-I. Kwon, "Model order reduction of nonlinear parabolic PDE systems with moving boundaries using sparse proper orthogonal decomposition: Application to hydraulic fracturing," *Comput. Chem. Eng.*, vol. 112, pp. 92–100, 2018.
- [23] J. O. Laguna, A. G. Olaya, and D. Borrajo, "A dynamic sliding window approach for activity recognition," in *Proc. Int. Conf. User Model., Adaptation, and Personalization*, 2011, pp. 219–230.
- [24] S. Van Vaerenbergh, J. Via, and I. Santamaría, "A sliding-window kernel RLS algorithm and its application to nonlinear channel identification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2006, vol. 5, pp. V–V.
- [25] M. Krallinger, M. Padron, and A. Valencia, "A sentence sliding window approach to extract protein annotations from biomedical articles," *BMC Bioinf.*, vol. 6, no. 1, 2005, Art. no. S19.
- [26] K.-L. Du and M. Swamy, "Radial basis function networks," in *Neural Networks and Statistical Learning*. New York, NY, USA: Springer, 2014, pp. 299–335.
- [27] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, 1991.
- [28] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
- [29] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B*, vol. 58, pp. 267–288, 1996.
- [30] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. New York, NY, USA: Oxford, 1997.
- [31] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—an updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, 2016.
- [32] K.-B. Yu, "Recursive updating the eigenvalue decomposition of a covariance matrix," *IEEE Trans. Signal Process.*, vol. 39, no. 5, pp. 1136–1145, May 1991.
- [33] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, no. 5, pp. 403–420, 1970.

- [34] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.
- [35] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1329–1338, Nov. 1996.
- [36] K. K. Ahn and H. P. H. Anh, "Inverse double NARX fuzzy modeling for system identification," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 1, pp. 136–148, Feb. 2010.
- [37] M. B. Christopher, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2016.
- [38] H.-G. Han, W. Lu, Y. Hou, and J.-F. Qiao, "An adaptive-PSO-based self-organizing RBF neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 104–117, 2016.



Bing-Chuan Wang received the B.E. degree in automation and the M.S. degree in control science and engineering from Central South University, Changsha, China, in 2013 and 2016, respectively. He is currently working toward the Ph.D. degree in the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong.

His current research interests include modeling of distributed parameter systems, battery management systems, and evolutionary

computation.



Han-Xiong Li (S'94–M'97–SM'00–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 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 broad experience in both academia and industry. He has authored 2 books and 7 patents, and published more than 200 SCI journal papers with h-index 41 (Web of Science). His current research interests include process modeling and control, system intelligence, distributed parameter systems, and battery management systems.

Dr. Li is currently an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and was an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS (2002–2016) and IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS (2009–2015). He was the recipient of the Distinguished Young Scholar (overseas) by the China National Science Foundation in 2004, a Chang Jiang professorship by the Ministry of Education, China, in 2006, and a national professorship in China Thousand Talents Program in 2010. He is currently a Distinguished Expert for Hunan Government and China Federation of Returned Overseas Chinese.