Incremental Spatiotemporal Learning for Online Modeling of Distributed Parameter Systems

Zhi Wang and Han-Xiong Li, *Fellow, IEEE*

***Abstract*—An incremental spatiotemporal learning scheme is proposed for online modeling of distributed parameter systems (DPSs). A novel incremental learning method is developed to recursively update the spatial basis functions and the corresponding temporal model based on the Karhunen–Loève decomposition for time-space separation. The time-space synthesis continually evolves by adding new increment data with more updated information and revising the existing parameters of the dynamic system. In this way, the spatiotemporal structure is inherited and updated efficiently as output data increases over time. The adaptive nature of this evolving structure makes it promising for online modeling of DPSs under streaming data environment. The proposed incremental modeling scheme is evaluated on the classical benchmark of a catalytic rod problem. The simulation results demonstrate the viability and efficiency of the proposed method for online modeling of DPSs.**

***Index Terms*—Distributed parameter systems (DPSs), incremental learning, Karhunen–Loève decomposition (KLD), online spatiotemporal modeling.**

I. Introduction

DISTRIBUTED parameter systems (DPSs) are a common kind of industrial processes where the input and output may vary in both time and space dimension [1]. Despite of the difficulty, modeling such complex systems is essential to industrial simulation, control, and optimization [2]–[4]. Modeling and control of such spatiotemporal systems has been widely investigated in practice due to recent developments in sensor, actuator, and computing technology. The first-principle description for known DPS conventionally leads to the mathematical partial differential equation (PDE). Since the PDE system is infinite-dimensional, the model reduction complements are always indispensable for real implementation.

The time-space separation methods have been verified to be an efficient model reduction method in modeling of unknown DPSs [5]–[11]. In these spatiotemporal modeling methods, Karhunen–Loève decomposition (KLD) is first utilized for the time-space separation, where the spatiotemporal output is decomposed into a set of dominant spatial basis functions (BFs) with corresponding temporal coefficients. Second, a reduced-order temporal model is identified from the decomposed low-dimensional data. The temporal structure can be approximated by various identification techniques, such as nonlinear autoregressive with exogenous input (NARX) model [12] Hammerstein model [5], neural networks (NNs) [7], [8], and so on. Finally, the spatiotemporal dynamics can be reconstructed and predicted over the whole time-space domain through the time-space pairwise data reconstruction of the reduced-order model.

In traditional spatiotemporal modeling, the KLD process and temporal structure identification are performed in the so-called batch-mode. The output data over the whole time domain has to be ready for time-space separation during the model training stage. The modeling procedure stops once the whole batch of spatiotemporal outputs has been fully processed. These methods assume that all the output data is available and accessible at the beginning of the modeling process. Therefore, they are feasible for offline implementations only. Nevertheless, in online settings, new streaming data will be available continually, even after the spatiotemporal model having been identified at a certain moment. If we want to incorporate additional new output data into the existing time-space synthesis, the time-space separation process should be restarted from scratch with all the new and the old training data, which is called as "batch-mode" shown in Fig. 1. Since the number of training data is growing constantly, the batch-mode method is only feasible at the cost of retraining the whole time-space synthesis with time-consuming procedures and great storage burden. Although some DPSs may have relatively slow dynamics, making such retraining scheme feasible. It is difficult to characterize it as adaptation, especially with respect to the model structure of the time-space synthesis. In fact, it is a procedure where completely new reduced-order models are repeatedly generated from scratch given the accumulated data with growing length.

From the aspect of computational efforts, calculating the Karhunen–Loève basis for L time steps of N spatial measurements requires roughly $O(NL)$ memory units and $O(L^3)$ flops. The growing data length L results in superlinearly increasing computational complexity and linearly increasing storage capacity for batch-mode method. In many real applications, this large storage requirements and computational demands may be prohibitive. Moreover, acquisition of representative training data is expensive and time-consuming. It is common

Manuscript received December 11, 2017; accepted February 14, 2018. Date of publication March 19, 2018; date of current version November 19, 2019. This work was supported by the General Research Fund Project from Research Grant Council of Hong Kong SAR under Grant CityU: 11205615. This paper was recommended by Associate Editor A. H. Tan. *(Corresponding author: Han-Xiong Li.)*

The authors are with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong (e-mail: njuwangzhi@gmail.com; mehxli@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/TSMC.2018.2810447

for such data available only in small increments over a period of time, and the previously visited data may be unaccessible in consideration of online storage. Under the circumstances, either we are not capable of collecting all the training data for time-space separation, or the time-space synthesis is identified from scratch inefficiently using all the data.

We can see that the batch calculation nature of the spatiotemporal modeling methods has limited their applications. It is an important obstacle in designing online modeling methods for distributed processes, since the traditional methods are still not adaptive. In turn, it is hard to scale up the developed modeling systems. An adaptive method of modeling DPSs is needed for online settings to overcome the above challenges of streaming data and computational limits. A new modeling scheme should be developed with the capability of evolving the time-space synthesis as the process continues on. The model structure is supposed to be inherited and updated whenever the new data increment is available. The new information carried by the new data should be added into the existing model structure in an incremental way, i.e., *incremental learning*.

Recent years have witnessed an increasing interest in the topic of incremental learning from both academia and industry. Incremental learning has been widely addressed in machine learning and intelligent control communities to cope with learning tasks, where the training data becomes available over time or the learning environment is ever-changing [13]. Various methods have been suggested for incremental learning regarding various problems in different areas, including unsupervised learning [14], supervised learning [15], reinforcement learning [16], machine vision [17], evolutionary algorithms [18], and human–robot interaction [19].

For model reduction of DPSs, there are several results reported regarding the concept of recursive, or adaptive, or incremental methods. Li *et al.* [20] proposed a recursive principle component analysis approach based on updating the correlation matrix recursively. Varshney *et al.* [21] and Pourkargar and Armaou [22], [23] developed a kind of adaptive proper orthogonal decomposition on the base of updating the BFs through orthonormalization of the dominant eigenspace of the covariance matrix. The algorithm requires the dimensionality of the covariance matrix to remain constant by discarding the oldest snapshots, which leads to a certain loss of the system's dynamics. Xu *et al.* [24] proposed a recursive proper orthogonal decomposition approach through gradient search of the new eigenspace, which aims at minimizing the approximation error. Sequently, they proposed a rank-1 incremental proper orthogonal decomposition method [25] and [26] based on expansion and transformation of the eigenspace by the normalized residue vector. These methods are mostly based on analysis of the covariance matrix, which requires access to all the historical data. Meantime, some of them may have deficiencies, such as information loss, limitation to rank-1 updating, and local minimum.

Regarding to spatiotemporal modeling of DPSs, there are few results reported concerning the incremental learning method, which is exactly needed for online modeling in environments of streaming data. Although there are several works proposing the concept of incremental modeling of DPSs [27], they refer in particular to adding the hierarchical spatiotemporal kernels incrementally, which is completely different from our proposed incremental algorithm for online modeling. The purpose of this paper is to present such incremental modeling methodology and results of their applications to a number of test cases.

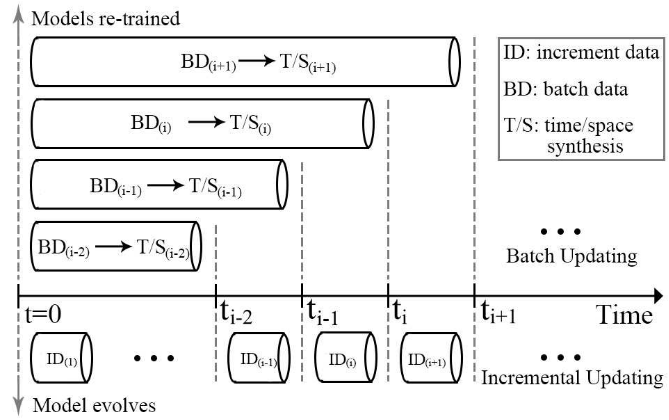


Fig. 1. Traditional batch-mode modeling versus proposed incremental-mode modeling.

The intuitive concept of proposed incremental learning methodology for online spatiotemporal modeling is as shown in Fig. 1, along with the comparison to conventional batch-mode modeling. In online settings, assume that the modeling procedure is processed continually at set time steps $(\ldots, t_{i-1}, t_i, t_{i+1}, \ldots)$. For the batch-mode method, completely new time-space syntheses $(\ldots, T/S_{(i-1)}, T/S_{(i)}, T/S_{(i+1)}, \ldots)$ are trained from scratch after collecting the whole batch data $(\ldots, \mathrm{BD}_{(i-1)}, \mathrm{BD}_{(i)}, \mathrm{BD}_{(i+1)}, \ldots)$. While for our proposed incremental modeling method, the time-space synthesis is inherited and updated in a computationally effective way by adding the new increment data $(\ldots, \mathrm{ID}_{(i-1)}, \mathrm{ID}_{(i)}, \mathrm{ID}_{(i+1)}, \ldots)$ into the existing model structure incrementally. It is not required to store the entire time series of training data before proceeding to the time-space separation, and this evolving structure is capable of approximating and adapting to the system's dynamics well in real-time.

In order to demonstrate the performances of the proposed incremental modeling algorithm, simulated experiments are carried out on the benchmark of a catalytic rod problem. We compare the incremental modeling algorithm with conventional batch-mode method to illustrate the feasibility and advantages of the incremental learning property. Both theoretical analysis and experimental results will demonstrate that the proposed incremental modeling methodology achieves good online performances, as well as being computationally effective.

The rest of this paper is organized as follows. In Section II, the problem description of online modeling is introduced. In Section III, we present the concept and technical details of the proposed incremental spatiotemporal learning scheme for online modeling, accompanied by illustrations of complexity analysis and main advantages. Experimental results are demonstrated in Section IV, and the conclusions are presented in Section V.

II. Problem Description

In this paper, a general class of DPSs is considered, which can be represented by the following nonlinear PDE:

$$\frac{\partial y(x,t)}{\partial t} = \mathcal{L}\left(y, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial x^2}, \ldots, \frac{\partial^{n_0} y}{\partial x^{n_0}}\right) + \bar{B}(x)u(t) \tag{1}$$

subject to the mixed-type boundary conditions

$$q\left(y, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial x^2}, \ldots, \frac{\partial^{n_0-1} y}{\partial x^{n_0-1}}\right)|_{x=x_a \text{ or } x=x_b} = 0 \tag{2}$$

and the initial condition

$$y(x,0) = y_0(x) \tag{3}$$

where $t \in [0, \infty)$ is the temporal variable, $x \in [x_a, x_b] \subset \mathbb{R}$ is the spatial coordinate, $y(x,t) = [y(x_1,t), \ldots, y(x_N,t)]^T \in \mathbb{R}^N$ is the spatiotemporal output, and $u(t) \in \mathbb{R}^p$ is the temporal input. $\mathcal{L} \in \mathbb{R}^N$ is a complex vector function which contains a nonlinear spatial differential operator of order n_0, $\bar{B}(x)$ is a matrix function of appropriate dimensions which describes how the temporal inputs are distributed in spatial domains, q is a nonlinear vector function, and $y_0(x)$ is a smooth vector function referring to the initial output.

A common approach to modeling the unknown nonlinear DPSs leads to the time-space separation framework [1], where the spatiotemporal output $y(x,t)$ can be decoupled into a set of orthogonal spatial BFs $\varphi(x)$ with corresponding temporal coefficients $a(t)$ as

$$y(x,t) = \sum_{i=1}^{\infty} \varphi_i(x) a_i(t). \tag{4}$$

In practice, a finite nth-order of BFs $\{\varphi_i(x)\}_{i=1}^n$ extracted by KLD is used for capturing the most relevant dynamics of the system. Then, the low-order temporal model $\mathcal{F}$ is identified from the decomposed low-dimensional coefficients $\{a_i(t)\}_{i=1}^n$ as

$$\begin{aligned} a(t) = \mathcal{F}(&a(t-1), \ldots, a(t-d_a), u(t-1), \ldots, u(t-d_u)) \\ &+ e(t) \end{aligned} \tag{5}$$

where d_u and d_a denote the maximum input and output lags, respectively, and $e(t)$ denotes the residual error. The detailed description of spatiotemporal modeling can be found in the Appendix.

Nevertheless, traditional spatiotemporal modeling methods are only feasible for offline implementations since the time-space synthesis is computed only once and remains fixed afterwards. In an online environment, the time-space synthesis is supposed to be retrained from scratch repeatedly when the new data is available, which leads to a high computational burden in real applications. For online modeling of DPSs, an incremental learning mechanism is needed to inherit and update the model structure efficiently whenever the new data increment is available.

III. Incremental Spatiotemporal Modeling

A. Framework

In the online environment, the output data for modeling is collected continuously, instead of being a fixed set. Some parts of the new collected data may confirm and reinforce the knowledge learned from the previous data; while other parts may bring new information that is sufficiently different from the learned knowledge, which could indicate complex dynamics such as abnormal interference or changes in operating conditions. Online methods are supposed to be adaptive to such dynamics of DPSs during their whole life cycles.

We present the technical details of the proposed incremental spatiotemporal modeling scheme in this section. The whole framework is shown in Fig. 2. The continuous streaming data is collected into data increments $(\ldots, \mathrm{ID}_{(i)}, \mathrm{ID}_{(i+1)}, \ldots)$ at certain time steps $(\ldots, t_i, t_{i+1}, \ldots)$. First, we propose an efficient method that incrementally updates the spatial BFs when a new data increment arrives. Second, the temporal model is reidentified using the corresponding updated temporal coefficients. Finally, we use the time-space synthesis $(\ldots, T/S_{(i)}, T/S_{(i+1)}, \ldots)$ with updated spatial BFs and temporal model to reconstruct the historical data $(\ldots, \hat{\mathrm{HD}}_{(i)}, \hat{\mathrm{HD}}_{(i+1)}, \ldots)$, and to predict the future outputs $(\ldots, \hat{\mathrm{ID}}_{(i+1)}, \hat{\mathrm{ID}}_{(i+2)}, \ldots)$. Then we repeat the above procedures whenever the next new increment of output data arrives. In this incremental way, the new increment data is added to the existing time-space synthesis continually. The modeling structure and parameters are inherited and updated recursively over time.

B. Online Updating of Time-Space Synthesis

Suppose that the output data at time $t_j (j = 1, \ldots, L)$ is an N-dimensional vector $y(x, t_j) = [y(x_1, t_j), \ldots, y(x_N, t_j)]^T$, which is measured at N spatial locations. For simplicity, mark $y_j = y(x, t_j)$. The n-order spatial BFs, denoted as $\{\varphi_i\}_{i=1}^n$, are typically learned by time-space separation from a set of training data $Y_1 = [y_1, \ldots, y_L]$ for time steps of L.

The output data is generated continually, even after the time-space synthesis has been learned at time step t_L. Suppose that the time-space synthesis should be processed at a new time step t_{L+M}, and $Y_2 = [y_{L+1}, \ldots, y_{L+M}]$ is the new data increment, for new time steps of M. For batch-mode method, the time-space separation is reperformed from scratch by KLD of the augmented data matrix $Y = [Y_1 \quad Y_2]$. This method is computationally expensive as the online process generates more and more historical data. Instead, we derive the concrete procedure on how the proposed method inherits and updates the time-space synthesis efficiently through incremental learning.

According to (31), the original temporal correlation matrix C can be written as

$$C = \frac{1}{L} Y_1^T Y_1. \tag{6}$$

By singular value decomposition (SVD), the matrix Y_1^T can be decomposed into

$$Y_1^T = U \Sigma V^T. \tag{7}$$

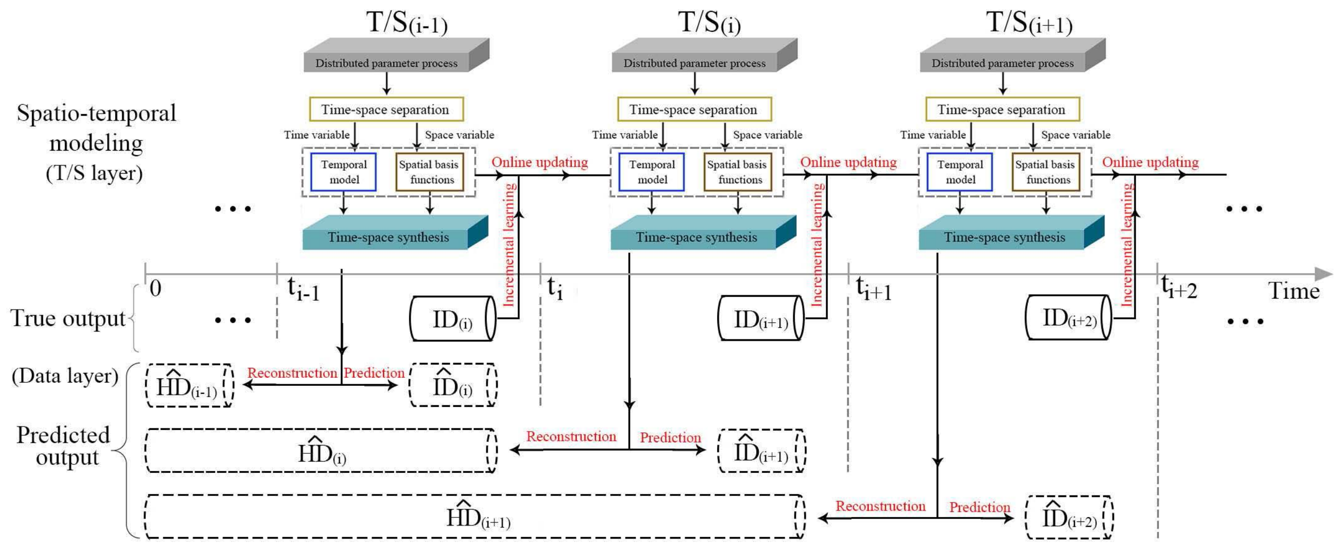


Fig. 2. Incremental spatiotemporal modeling scheme for online modeling of DPSs.

Then C can be rewritten as

$$C = \frac{1}{L} U \Sigma V^T V \Sigma U^T = \frac{1}{L} U \Sigma \Sigma^T U^T = U \Lambda U^T \tag{8}$$

where $\Lambda = (1/L)\Sigma\Sigma^T$ is an $L \times L$ diagonal matrix. By KLD, we choose the dominant n features which capture more than 99% of the system's information according to (32). Then the best rank-n approximation of C is

$$C_n = U_n \Lambda_n U_n^T \tag{9}$$

where U_n is formed by the first n columns of U, and Λ_n is the nth leading principal submatrix of Λ. According to (28), we can construct the n dominant BFs $\Phi = [\varphi_1(x), \ldots, \varphi_n(x)]$ as

$$\Phi = \left(U_n^T Y_1^T\right)^T = Y_1 U_n. \tag{10}$$

After identifying the dominant spatial BFs $\{\varphi_i(x)\}_{i=1}^n$, the corresponding temporal coefficients $\{a_i(t)\}_{i=1,t=1}^{n,L}$ of the spatiotemporal output $y(x,t)$ can be obtained using (23). Assume that the acquired temporal coefficients matrix is $A_{n\times L} = [a(1), \ldots, a(L)]$, where $a(t) = [a_1(t), \ldots, a_n(t)]^T, t = 1, \ldots, L$, it can be verified that the output data Y_1 is reconstructed, as $(\widehat{Y}_1)_n$, using spatial BFs Φ and the corresponding temporal coefficients A

$$\left(\widehat{Y}_1\right)_n = \Phi A. \tag{11}$$

When the new data Y_2 is added and $Y = [Y_1, Y_2]$, the new temporal correlation matrix $\bar{C}$ is

$$\bar{C} = \frac{1}{L+M} Y^T Y = \frac{1}{L+M} \begin{bmatrix} Y_1^T Y_1 & Y_1^T Y_2 \\ Y_2^T Y_1 & Y_2^T Y_2 \end{bmatrix}. \tag{12}$$

Suppose that the previous data Y_1 is not accessible any more, the new $\bar{C}$ cannot be computed directly. Instead, we update the eigenvectors of data matrix Y to compute the new BFs based on the SVD-updating algorithm [37], [38] in an incremental way.

As we known, $Y_1^T \in \mathbb{R}^{L\times N}$, $(Y_1^T)_{L\times N} = U\Sigma V^T$, and its best rank-$n$ approximation $(\widehat{Y}_1^T)_n = U_n \Sigma_n V_n^T$, where U_n and V_n are formed by the first n columns of U and V, respectively, and Σ_n is the nth leading principal submatrix of Σ. Next, we want to carry out the SVD of a larger matrix $\begin{bmatrix} (Y_1^T)_{L\times N} \\ (Y_2^T)_{M\times N} \end{bmatrix}$, where Y_2^T is an $M \times N$ matrix consisting of M additional rows.

Let the QR decomposition of $(I - V_n V_n^T)Y_2$ be

$$\left(I - V_n V_n^T\right) Y_2 = \text{QR} \tag{13}$$

where Q is orthonormal and R is the $m \times M$ upper triangular, where m ($m \le \min(N, M)$) is the rank of $(I - V_n V_n^T)Y_2$. This step projects the new rows Y_2^T to the orthogonal complement of the old right eigenvector subspace, i.e., span$\{V_n\}$. It can be verified that

$$Y^T = \begin{bmatrix} \left(Y_1^T\right)_{L\times N} \\ \left(Y_2^T\right)_{M\times N} \end{bmatrix} = \begin{bmatrix} U_n & 0 \\ 0 & I_M \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ Y_2^T V_n & R^T \end{bmatrix} \begin{bmatrix} V_n & Q \end{bmatrix}^T \tag{14}$$

noticing that $[V_n \quad Q]$ is orthonormal. Now, obtain the SVD of the $(n+M)\times(n+m)$ matrix

$$\begin{bmatrix} \Sigma_n & 0 \\ Y_2^T V_n & R^T \end{bmatrix} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T \tag{15}$$

where $\widetilde{U} \in \mathbb{R}^{(n+M)\times(n+M)}$ and $\widetilde{\Sigma} \in \mathbb{R}^{(n+M)\times(n+m)}$, and $\widetilde{V} \in \mathbb{R}^{(n+m)\times(n+m)}$.

Then, the new temporal correlation matrix $\bar{C}$ can be rewritten as

$$\begin{aligned} \bar{C} &= \frac{1}{L+M} Y^T Y \\ &= \frac{1}{L+M} \begin{bmatrix} U_n & 0 \\ 0 & I_M \end{bmatrix} \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T \begin{bmatrix} V_n & Q \end{bmatrix}^T \\ &\quad \times \begin{bmatrix} V_n & Q \end{bmatrix} \widetilde{V}\widetilde{\Sigma}^T\widetilde{U}^T \begin{bmatrix} U_n & 0 \\ 0 & I_M \end{bmatrix}^T \\ &= \left(\begin{bmatrix} U_n & 0 \\ 0 & I_M \end{bmatrix}\widetilde{U}\right)\left(\frac{1}{L+M}\widetilde{\Sigma}\widetilde{\Sigma}^T\right)\left(\begin{bmatrix} U_n & 0 \\ 0 & I_M \end{bmatrix}\widetilde{U}\right)^T \\ &= \bar{U}\bar{\Lambda}\bar{U}^T \end{aligned} \tag{16}$$

where the updated diagonal matrix $\bar{\Lambda} = (1/L+M)\widetilde{\Sigma}\widetilde{\Sigma}^T$, and the updated eigenvectors $\bar{U} = \begin{bmatrix} U_n & 0 \\ 0 & I_M \end{bmatrix}\widetilde{U}$. By KLD, we choose the new dominant n' features which capture more than 99% of the system's information according to (32). The best rank-n' approximation of $\bar{C}$ is

$$\bar{C}_{n'} = \bar{U}_{n'}\bar{\Lambda}_{n'}\bar{U}_{n'}^T \tag{17}$$

where $\bar{U}_{n'}$ is formed by the first n' columns of $\bar{U}$, and $\bar{\Lambda}_{n'}$ is the n'th leading principal submatrix of $\bar{\Lambda}$.

In accordance with (28), we can update the previous n-order BFs $\Phi = [\varphi_1(x), \ldots, \varphi_n(x)]$ to a new n'-order one as $\bar{\Phi} = [\bar{\varphi}_1(x), \ldots, \bar{\varphi}_{n'}(x)]$

$$\bar{\Phi} = \left(\bar{U}_{n'}^T Y^T\right)^T = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix}\bar{U}_{n'}. \tag{18}$$

Since the complete information about the original data Y_1 is not accessible due to online storage, we use the best rank-n approximation (11) to reconstruct the original data. Then the new n'-order dominant BFs can be computed as

$$\bar{\Phi} = \begin{bmatrix} \Phi A & Y_2 \end{bmatrix}\left(\begin{bmatrix} U_n & 0 \\ 0 & I_M \end{bmatrix}\widetilde{U}_{n'}\right) \tag{19}$$

where $\widetilde{U}_{n'}$ is formed by the first n' columns of $\widetilde{U}$. In this incremental way, the old BFs Φ is transformed to an updated one $\bar{\Phi}$ when the new increment of output data Y_2 arrives, without requirement to store the previous data Y_1. This function enables recursive calculation, which is important for online implementation of modeling methods.

After the spatial BFs being updated, the corresponding temporal coefficients can be updated according to (34), following by reidentification of the low-order temporal model in (35). At this point, the whole time-space synthesis has been inherited and updated online for reconstructing the system dynamics and predicting future outputs in real-time. By incremental learning, the modeling structure evolves continually as new increments of spatial measurements are generated through the whole life cycle of DPSs. Hence, this evolving structure is capable of tracking and adapting to the system's dynamics online.

C. Computational Complexity of Incremental Modeling

The first step of the proposed incremental modeling is QR decomposition of $[(I - V_nV_n^T)Y_2]_{N\times M}$ in (13), which requires approximately $O(NM^2)$ flops. The following is SVD of the smaller matrix $\begin{bmatrix} \Sigma_n & 0 \\ Y_2^T V_n & R^T \end{bmatrix}_{(n+M)\times(n+m)}$ that requires approximately $O((n+m)(n+M)^2)$ flops. In many applications, the number of dominant BFs n is much smaller than other parameters. That is to say, $n \ll \{m, N, M\}$, and $m \leq \min\{N, M\}$. Neglecting the contribution of initialization step, the total time complexity of the incremental learning procedure is at the level of $O(NM^2)$, depending on the length M of the new data increment. Nevertheless, in batch-mode method, it requires approximately $O((L+M)^3)$ flops to proceed the KLD of the new correlation matrix $\bar{C}_{(L+M)\times(L+M)}$. Hence, the computational complexity of incremental modeling will be much lower than the batch-mode method, since the historical data length L in online mode is continuously growing, which leads to $\{N, M\} \ll L$ in many practical cases.

D. Main Advantages of Incremental Modeling

1) *Online Computation and Database Update:* It deals with a continuous sequence of spatial measurements, processes the streaming data as it arrives in real-time rather than waiting for the end of the sequence, without any requirement to keep the previous measurements as well.
2) *Reduced Complexity and Memory Requirements:* The incremental learning process requires approximately $O(NM^2)$ flops and $O(NM)$ memory units in comparison to $O((L+M)^3)$ flops and $O(N(L+M))$ memory units required by the batch-mode method.
3) *Adaptiveness:* It develops a continually inherited and updated time-space synthesis according to new increments of output data, which can track and adapt to the system's dynamics in real-time.

IV. Simulation Experiments

In order to evaluate the proposed incremental spatiotemporal modeling methodology, the benchmarked distributed process of a catalytic rod is studied. At each time step t_j, the spatial measurements vector $y_j = y(x, t_j)$ is acquired. Suppose that during the time period between step t_L and t_{L+M}, we collect the new increment of output data $Y_2 = [y_{L+1}, \ldots, y_{L+M}]$. The existing time-space synthesis learned from the historical data $Y_1 = [y_1, \ldots, y_L]$ at t_L shall be transformed to an updated version at t_{L+M} through incremental learning of the new data set Y_2. The up-to-date synthesis is used to reconstruct the system's output and predict the system's dynamics in the future. Then, the updating process is repeated whenever the next new increment of output data arrives. In this incremental way, the time-space synthesis is inherited and updated recursively, resulting in implementation of online modeling and prediction in real-time.

In order to demonstrate the performance of our proposed incremental modeling methodology, we compare it to the traditional batch-mode modeling method. In batch-mode, the spatial BFs and temporal coefficients are computed directly using all the training data from the initial state to the present of the process, assuming that the previous data Y_1 was accessible. All the algorithms are implemented in MATLAB R2013a running on Windows 7 with Intel core i5-4590 3.30 GHz and 4 GB RAM. And all the experimental results presented in this paper are averaged over 100 runs.

Let $y(x, t)$ and $y_n(x, t)$ denote the measured output and the predicted output. The three performance indexes for evaluating the modeling accuracy is defined as follows.

1) Spatiotemporal error $e(x, t) = y(x, t) - y_n(x, t)$.
2) Spatial normalized absolute error, $\text{SNAE}(t) = (1/N)\sum_{i=1}^{N}|e(x_i, t)|$.
3) Root of mean squared error, $\text{RMSE} = (\int\sum e(x, t)^2dx/\int dx\sum\Delta t)^{1/2}$.

A. Case: Catalytic Rod

The benchmark PDE system of a catalytic rod, which consists a long thin rod in a reactor, is shown in Fig. 3. It is a classical and widely investigated transport-reaction process in chemical industry [39]. A zeroth-order exothermic chemical

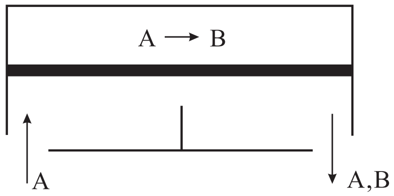


Fig. 3. Catalytic rod.

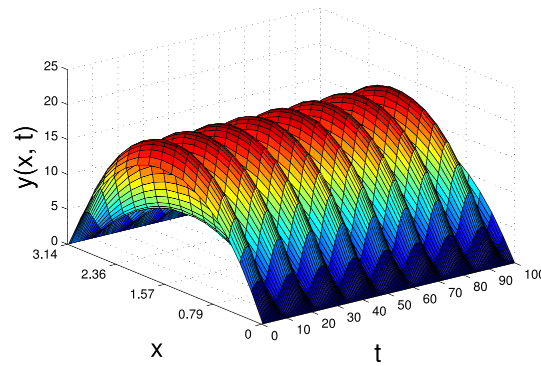


Fig. 4. Measured output for a period of the online process.

reaction is produced inside in the form of $A \to B$, where A is the pure species fed into the reactor. A cooling medium in touch with the catalytic rod is used for cooling the exothermic process.

Assume the species A in the furnace is excess, and the following parameters of the catalytic rod are constant: density, heat capacity, conductivity, and temperature at both sides. The mathematical model of the following parabolic PDE can be used to describe the spatiotemporal evolution of the rod temperature [39]:

$$\frac{\partial y(x,t)}{\partial t} = \frac{\partial^2 y(x,t)}{\partial x^2} + \beta_T\left(e^{-\frac{\gamma}{1+y}} - e^{-\gamma}\right) + \beta_u\left(b^T(x)u(t) - y(x,t)\right) \quad (20)$$

subject to the Dirichlet boundary and initial conditions

$$y(0,t) = 0, \qquad y(\pi,t) = 0, \qquad y(x,0) = y_0(x)$$

where $y(x,t)$ is the rod temperature, $u(t)$ is the temporal input function, and $b(x)$ is the spatial distribution of input actuators. β_T is the heat of reaction, β_u is the heat transfer coefficient, and γ denotes the activation energy. The process parameters are often set as

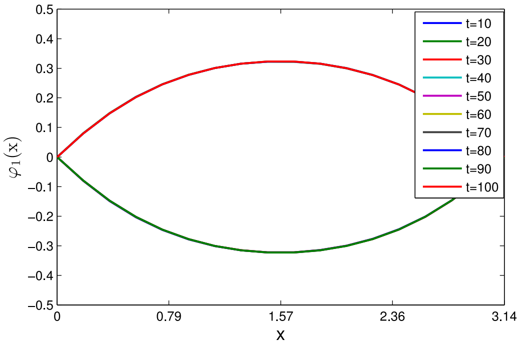
$$\beta_T = 50, \qquad \beta_u = 2, \qquad \gamma = 4.$$

There are four input actuators $u(t) = [u_1(t), \ldots, u_4(t)]^T$ with the spatial distribution function $b(x) = [b_1(x), \ldots, b_4(x)]^T$, $b_i(x) = H(x-(i-1)\pi/4) - H(x-i\pi/4)$, $(i = 1, \ldots, 4)$ and $H(\cdot)$ is the standard Heaviside function. For gathering informative data and persistently exciting the full spectrum of the nonlinear system's dynamics, the input signals are implemented with a series of sinusoidal functions with different frequencies as $u_i(t) = 1.1 + 5\sin(t/2 + i/10)$, $(i = 1, \ldots, 4)$. The number of required sensors for modeling depends on both the intrinsic physical system and the extrinsic modeling accuracy needed in practice. In this case, the system's output $y(x_i, t)$, $(i = 1, \ldots, N)$ is collected from 18 identical sensors that are uniformly distributed in the spatial domain $(N = 18)$.

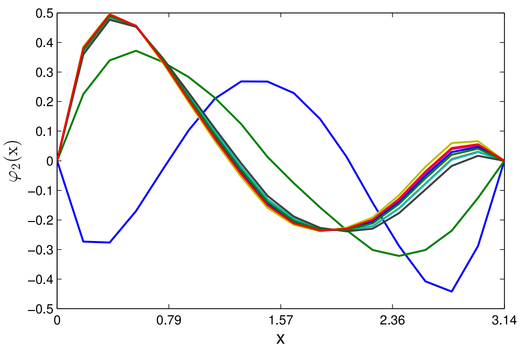
The noise-free streaming data is generated from (20) continually, which is sampled at time interval $\Delta t = 0.01$. The initial condition $y_0(x)$ is set to be the steady state with the input $u_i(t) = 1.1$, $(i = 1, \ldots, 4)$. The white Gaussian noise with mean zero and standard deviation $\sigma(x_i) = A_d(x_i)n_d$, where $A_d(x_i) = (\max(y(x_i,t)) - \min(y(x_i,t)))/3$, $(i = 1, \ldots, N)$ and $n_d = 2\%$ is added additively to the noise-free streaming data to derive the noisy output. The streaming output data is collected for updating the time-space synthesis at time interval $\Delta_c t = 10$. That is, the original spatial BFs and the temporal model is computed when the first 1000 output data is collected at time $t = 10$. Then the new data is added to the existing time-space synthesis in increments of 1000 at time $t = 20, 30, \ldots$ Subsequently, the time-space synthesis is inherited and updated through incremental learning every time the next new 1000 data is collected. In these moments, the up-to-date time-space synthesis is used for reconstructing the system's output from the initial state to the present. And for verifying the online modeling performance. It is used to predict the 1000 output data during the future time interval $\Delta_c(t)$. In this way, the incremental spatiotemporal model is trained and tested online in real-time, which is capable of being adaptive to the system's dynamics.

As a short example, the measured output $y(x,t)$ for $t \in (0, 100)$ is shown in Fig. 4. In the experiment, three dominant spatial BFs are selected since they can capture more than 99% of the system's energy all the time. As shown in Fig. 5, the three BFs $\{\varphi_i(x)\}_{i=1}^3$ are updated every $\Delta_c t = 10$. It can be observed that the first spatial basis oscillates between two sets of values, while the changes of both the second and the third basis are getting smaller and smaller along with the online process.

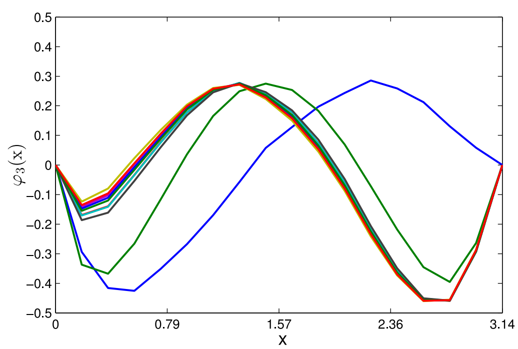
For intuitive comparison, we use RMSE as performance index regarding to modeling accuracy of the proposed incremental modeling and the traditional batch-mode method. At each time step when the new 1000 data is added in $(t = 20, 30, \ldots)$, we compute the RMSE on the training data from the initial state to the present, and RMSE on the 1000 testing data during the future time interval $\Delta_c(t)$. We also compute the modeling errors when the BFs are not updated all the way to illustrate the necessity of updating the BFs in the online environment. Together, the modeling accuracy comparison is shown in Table I. From the table, we can clearly see that the reconstruction error on the training data of the incremental method has always been very close to the batch-mode method. By calculation, the difference of the reconstruction error between these two methods is less than 1%. Meantime the testing errors for predicting the future outputs of the two methods are equally comparable to each other. These two indexes indicate that the incremental modeling method achieves almost as good performance as the traditional batch-mode method in terms of modeling accuracy.



(a)



(b)



(c)

Fig. 5. First three dominant BFs derived by the proposed incremental modeling. (a) $\varphi_1(x)$. (b) $\varphi_2(x)$. (c) $\varphi_3(x)$.

TABLE I
MODELING ACCURACY COMPARISON BETWEEN THE TRADITIONAL BATCH UPDATING AND THE PROPOSED INCREMENTAL UPDATING METHODS

Without updating / Batch updating / Incremental updating		
Time	RMSE training	RMSE testing
t = 20	0.0788 / 0.0433 / 0.0434	0.0844 / 0.0550 / 0.0549
t = 30	0.0807 / 0.0434 / 0.0435	0.0834 / 0.0555 / 0.0553
t = 40	0.0814 / 0.0435 / 0.0436	0.0834 / 0.0552 / 0.0551
t = 50	0.0818 / 0.0435 / 0.0436	0.0818 / 0.0552 / 0.0549
t = 60	0.0818 / 0.0435 / 0.0436	0.0850 / 0.0551 / 0.0550
t = 70	0.0823 / 0.0435 / 0.0437	0.0841 / 0.0555 / 0.0557
t = 80	0.0825 / 0.0435 / 0.0437	0.0832 / 0.0559 / 0.0555
t = 90	0.0826 / 0.0435 / 0.0437	0.0835 / 0.0556 / 0.0554
t = 100	0.0827 / 0.0435 / 0.0437	0.0832 / 0.0553 / 0.0549

On the other hand, the running time for updating the time-space synthesis is considered as the performance index for evaluating the computational efficiency. As shown in Fig. 6,

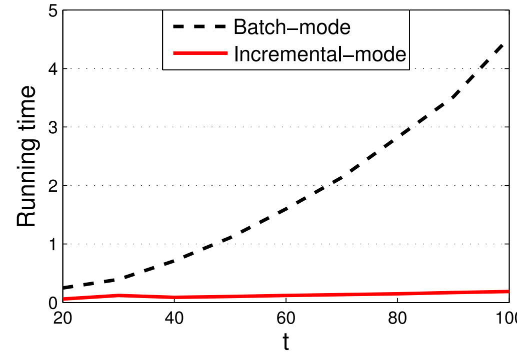


Fig. 6. Running time (s) comparison between the traditional batch-mode modeling and the proposed incremental-mode modeling.

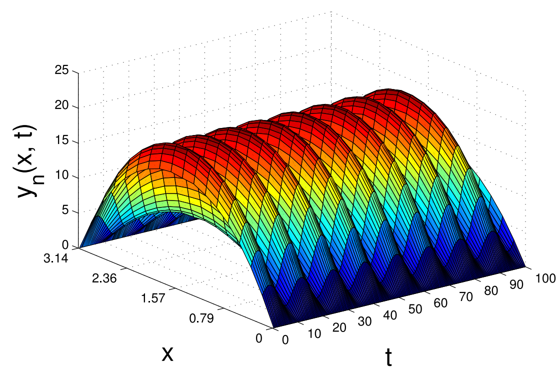


Fig. 7. Predicted output of incremental modeling on training data.

the running time of the batch-mode method increases super-linearly over time, since its time complexity is $O((L+M)^3)$ as the online process resulting in a growing number of historical output data with length L. Nevertheless, the running time of the incremental modeling increases very slowly. This attracting advantage should benefit from its time complexity being $O(NM^2)$, which depends on the data increment length M instead of the historical data length L. This index shows that the incremental modeling is computationally much more effective than the batch-mode method.

For more intuitive performance demonstration and contrast of model training, we present the predicted output $y_n(x,t)$, the spatiotemporal error $e(x,t)$ and spatial normalized absolute error SNAE(t) on the training data $t \in (0, 100)$, as shown in Figs. 7–9, respectively, Similarly, for further verifying the performance of model testing, the measured output $y(x,t)$, the predicted output $y_n(x,t)$, the spatiotemporal error $e(x,t)$, and spatial normalized absolute error SNAE(t) on a new set of 2000 testing data are also illustrated in Figs. 10–13, respectively. Obviously, it can be found that the proposed incremental modeling performs equally good as the traditional batch-mode method, and can provide an extremely close approximation to the original system.

Combined with the theoretical analysis in Section III, it can be summarized that the expected modeling accuracy and computational gains are indeed achieved. From the perspective of modeling accuracy, the proposed incremental method on inheriting and updating the time-space synthesis gives an

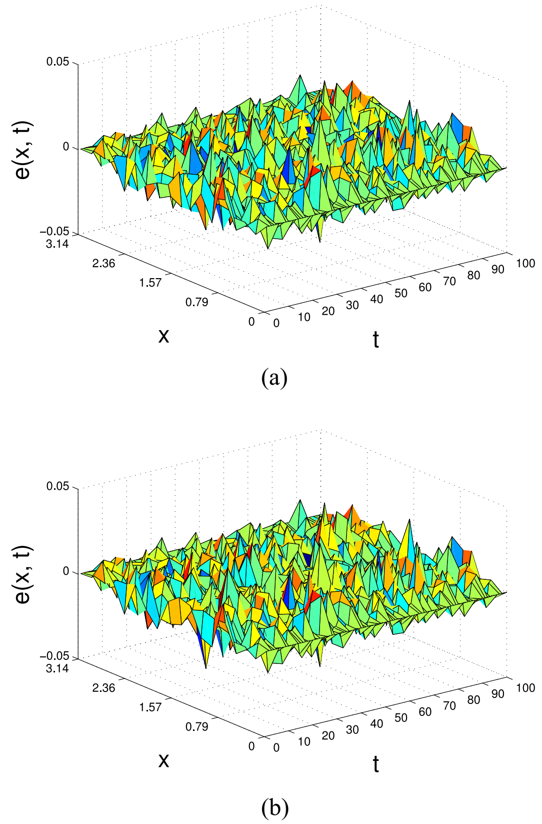


Fig. 8. Comparison on spatiotemporal error of (a) incremental-mode and (b) batch-mode methods on training data.

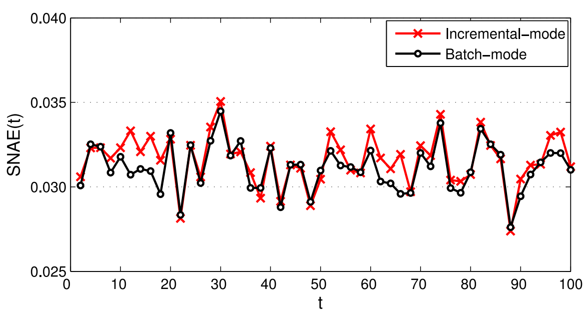


Fig. 9. Comparison on spatial normalized absolute error of incremental-mode and batch-mode methods on training data.

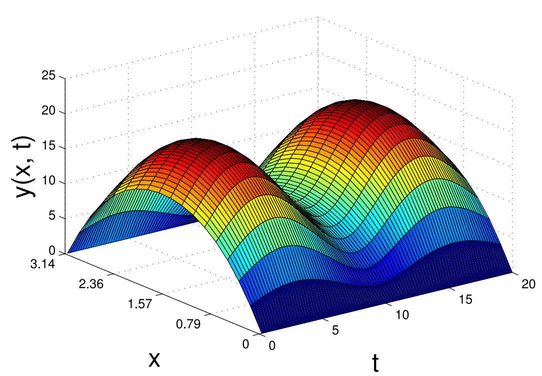


Fig. 10. Measured output for the testing.

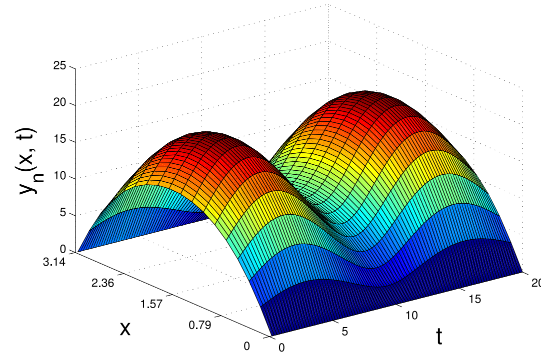


Fig. 11. Predicted output of incremental modeling on testing data.

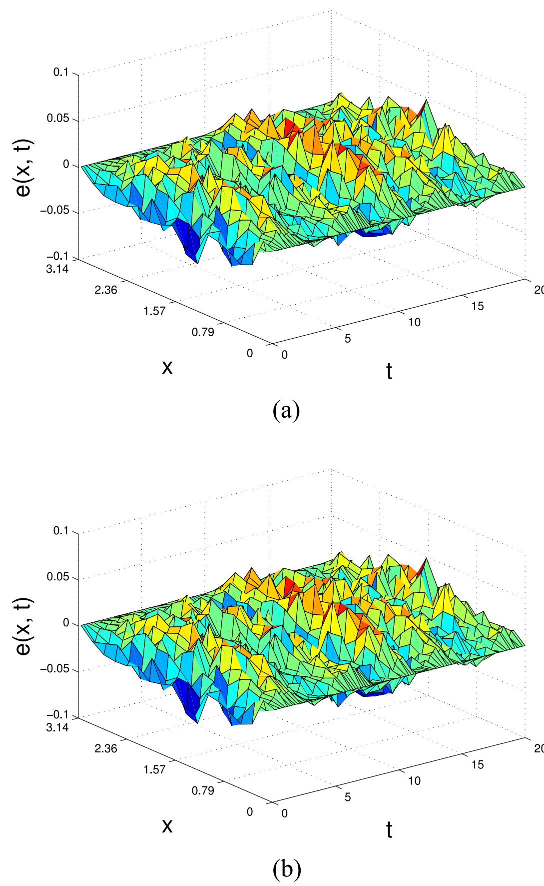


Fig. 12. Comparison on spatiotemporal error of (a) incremental-mode and (b) batch-mode methods on testing data.

extremely close approximation to the traditional batch-mode method. At the same time, the proposed incremental modeling has the advantages of saving much computational effort, being adaptive to online processes, and no requirement to store the previous data.

Remark 1: The time interval for updating the time-space synthesis, denoted as $\Delta_c(t)$, is a hyperparameter in the incremental modeling algorithm. In the experiment, we evaluate the model training and testing performances of the incremental modeling algorithm at time $t = 100$ with respect to different settings of updating time interval. As shown in Fig. 14, it can be observed that the incremental modeling algorithm achieves almost the same good performances regarding to different updating time intervals. In practice, the updating time interval can be adjusted according to the process requirements.

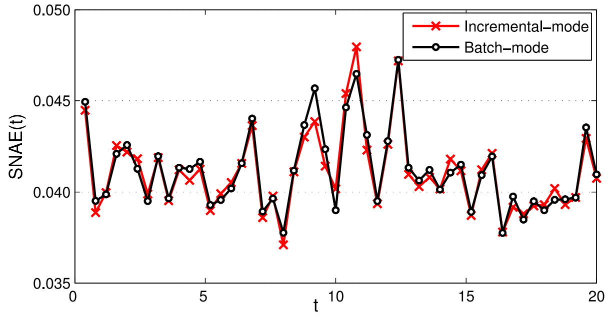


Fig. 13. Comparison on spatial normalized absolute error of incremental-mode and batch-mode methods on testing data.

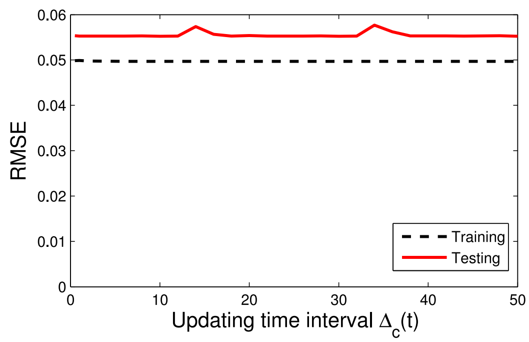


Fig. 14. Modeling error of the proposed incremental algorithm with respect to the updating time interval.

V. Conclusion

An incremental spatiotemporal learning scheme is proposed for online modeling of DPSs in this paper. It is based on recursive updating of the spatial BFs and the corresponding temporal model through incremental learning of new sets from streaming data. In this way, the time-space synthesis is inherited and updated through the whole life cycle of the online process. The proposed incremental method can achieve almost the same modeling accuracy as the traditional batch-mode method. Meantime, it is computationally much more effective, since it does not require to retrain the whole model structure from scratch when new output data arrives. Besides, it does not require to store the entire set of the process data. The adaptive nature of this methodology makes it promising for online modeling of DPSs for the whole life cycle. The proposed concept of incremental learning will have broad applications in many fields, including modeling, optimal sensor placement, and predictive control of DPSs. Experimental results demonstrate the viability, efficiency, and potential of this incremental-mode approach for online modeling of distributed processes. Future implementation in various engineering problems is under consideration.

Appendix
Spatiotemporal Modeling

A. Time-Space Separation

For time-space separation of the PDE system (1), KLD [28]–[29], as a data-based model reduction method for representing a stochastic field with the lowest dimension, is widely utilized for calculating the empirical eigenfunctions and deriving accurate reduced-order approximations of many PDE systems [5]–[11]. For simplicity, assume the system output $\{y(x_i, t)\}_{i=1,t=1}^{N,L}$, denoted as "snapshots," is uniformly sampled in both the time and space coordinates, where L is the time length. Define the inner product, norm and ensemble average as $(f_1(x), f_2(x)) = \int_\Omega f_1(x)f_2(x)dx$, $||f_1(x)|| = (f_1(x), f_1(x))^{1/2}$ and $\langle f_1(x,t)\rangle = (1/L)\sum_{t=1}^{L} f_1(x,t)$.

Motivated by Fourier series, the spatiotemporal variable $y(x,t)$ can be expanded onto an infinite number of orthonormal spatial BFs $\{\varphi_i(x)\}_{i=1}^{\infty}$ with temporal coefficients $\{a_i(t)\}_{i=1}^{\infty}$

$$y(x,t) = \sum_{i=1}^{\infty} \varphi_i(x)a_i(t). \tag{21}$$

Because the spatial BFs are orthonormal, i.e.,

$$\left(\varphi_i(x), \varphi_j(x)\right) = \int_\Omega \varphi_i(x)\varphi_j(x)dx = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \tag{22}$$

the temporal coefficients can be obtained from

$$a_i(t) = (\varphi_i(x), y(x,t)), i = 1, \ldots, \infty. \tag{23}$$

For practical use, it can be truncated into a finite-dimensional version

$$y_n(x,t) = \sum_{i=1}^{n} \varphi_i(x)a_i(t) \tag{24}$$

where $y_n(x,t)$ denotes the nth-order approximation.

Time-space separation aims to compute the most dominant spatial BFs $\{\varphi_i(x)\}_{i=1}^{n}$ among the spatiotemporal output $\{y(x_i,t)\}_{i=1,t=1}^{N,L}$ using KLD. Finding the typical $\{\varphi_i(x)\}_{i=1}^{n}$ can be performed by minimizing the following objective function:

$$\min_{\varphi_i(x)} \langle \|y(x,t) - y_n(x,t)\|^2 \rangle \tag{25}$$

subject to $(\varphi_i, \varphi_i) = 1, \varphi_i \in L^2(\Omega), i = 1, \ldots, n$. The orthonormal constraint $(\varphi_i, \varphi_i) = 1$ is imposed to restrict that the function $\varphi_i(x)$ is unique. The Lagrangian function with regard to this constrained optimization problem is

$$J = \langle \|y(x,t) - y_n(x,t)\|^2 \rangle + \sum_{i=1}^{n} \lambda_i((\varphi_i, \varphi_i) - 1) \tag{26}$$

the necessary condition of this problem can be computed as

$$\int_\Omega R(x,\xi)\varphi_i(\xi)d\zeta = \lambda_i\varphi_i(x), (\varphi_i, \varphi_i) = 1, i = 1, \ldots, n \tag{27}$$

where $R(x,\xi) = \langle y(x,t)y(\xi,t)\rangle$ is denoted as the spatial two-point correlation function.

The solution of (27) can be obtained by a computationally efficient method of snapshots [28]. The eigenfunction (spatial BFs) $\varphi_i(x)$ can be transformed into a linear combination of the snapshots as

$$\varphi_i(x) = \sum_{t=1}^{L} \gamma_{it} y(x,t). \tag{28}$$

After substituting (28) into (27), the necessary condition is computed as

$$\int_{\Omega} \frac{1}{L} \sum_{t=1}^{L} y(x,t)y(\zeta,t) \sum_{k=1}^{L} \gamma_{ik} y(\zeta,k) d\zeta = \lambda_i \sum_{t=1}^{L} \gamma_{it} y(x,t). \tag{29}$$

Then this eigenvalue problem is transformed to a simplified form of an $L \times L$ matrix eigen-decomposition problem as

$$C\gamma_i = \lambda_i \gamma_i \tag{30}$$

where $\gamma_i = [\gamma_{i1}, \ldots, \gamma_{iL}]^T$ is the ith eigenvector, and

$$C_{tk} = \frac{1}{L} \int_{\Omega} y(\zeta,t)y(\zeta,k)d\zeta \tag{31}$$

is defined as the temporal two-point correlation function. The solution of problem (30) yields the eigenvectors $\gamma_1, \ldots, \gamma_L$, which in turn can be used for constructing the eigenfunctions $\varphi_1(x), \ldots, \varphi_L(x)$ in (28). Since the matrix C is symmetric and positive semidefinite, the computed eigenfunctions are orthogonal.

Denote the maximum number of nonzero eigenvalues as $K \leq \min(N, L)$. Let the eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_K$ and the corresponding eigenfunctions $\varphi_1(x), \varphi_2(x), \ldots, \varphi_K(x)$ in the descending order of the magnitude of the eigenvalues. The eigenfunction corresponding to the first eigenvalue is supposed to be the most "energetic." The total "energy" of the PDE system is considered as the sum of the eigenvalues. The energy percentage to each eigenfunction based on the associated eigenvalue is assigned as

$$E_i = \lambda_i / \sum_{j=1}^{K} \lambda_j, i = 1, \ldots, K. \tag{32}$$

Usually, the sufficient set of eigenfunctions that capture more than 99% of the system's energy can be used to determine the reduced-order degree of n in (24). By experience, only a small set of dominant spatial BFs expansion can approximate most of the dynamics of many spatiotemporal systems. For any arbitrary set of spatial BFs $\{\phi_i(x)\}_{i=1}^n$, the following result holds [30]:

$$\sum_{i=1}^{n} \langle (y(\cdot,t), \varphi_i)^2 \rangle = \sum_{i=1}^{n} \lambda_i \geq \sum_{i=1}^{n} \langle (y(\cdot,t), \phi_i)^2 \rangle. \tag{33}$$

It shows that KLD is optimal on average in the class of representations by linear combination. That is why KLD can provide the lowest dimension n.

B. Temporal Model Identification

After learning the optimal spatial BFs $\{\varphi_i(x)\}_{i=1}^n$ by time-space separation, the low-order temporal model $a_i(t)$ is identified from the decomposed low-dimensional data. The temporal coefficients $a_i(t)$ corresponding to the spatiotemporal output $y(x,t)$ are computed from (21) as

$$a_i(t) = (\varphi_i(x), y(x,t)), i = 1, \ldots, n. \tag{34}$$

The time series data $a(t)$ is usually approximated by a deterministic NARX model [31]

$$a(t) = \mathcal{F}(a(t-1), \ldots, a(t-d_a), u(t-1), \ldots, u(t-d_u)) + e(t) \tag{35}$$

where d_u and d_a denote the maximum input and output lags, respectively, and $e(t)$ denotes the residual error. The unknown function $\mathcal{F}$ can be estimated from the low-dimensional input–output data set $\{u(t), a(t)\}_{t=1}^L$ using various function approximators, such as radial BFs (RBFs), polynomial functions, wavelets and kernel functions [32]. After identification, the model (35) can provide a prediction $\hat{a}(t)$ at any time t if the initial conditions are given. Combined with (24), this reduced-order model can reconstruct and predict the spatiotemporal dynamics over the entire time-space domain.

In this paper, the temporal model is assumed to be a simplified form as

$$a(t) = Ba(t-1) + \bar{\mathcal{F}}(a(t-1)) + Du(t-1) + e(t) \tag{36}$$

where the matrices $B \in \mathbb{R}^{n \times n}$ and $D \in \mathbb{R}^{n \times m}$ donate the linear part and the transform function $\bar{\mathcal{F}} : \mathbb{R}^n \to \mathbb{R}^n$ donates the nonlinear part. NNs are capable of approximating any continuous function to an arbitrary accuracy and have been widely investigated for various industrial processes [7], [8], [33]–[36]. In the temporal identification stage, $\bar{\mathcal{F}}$ is estimated as an RBF network, then the model (36) is rewritten as

$$a(t) = Ba(t-1) + WK(a(t-1)) + Du(t-1) + e(t) \tag{37}$$

where $W = [W_1, \ldots, W_l] \in \mathbb{R}^{n \times l}$ denotes the weight, $K(\cdot) = [K_1(\cdot), \ldots, K_l(\cdot)]^T : \mathbb{R}^n \to \mathbb{R}^l$ denotes RBF, and l is the number of neurons. The RBF is usually selected as the Gaussian kernel $K_i(a) = \exp-(a - c_i)^T \Sigma_i^{-1}(a - c_i)/2, (i = 1, \ldots, l)$ with proper center vector $c_i \in \mathbb{R}^n$ and norm matrix $\Sigma_i \in \mathbb{R}^{n \times n}$. With the KLD as a preprocessor, the size of the temporal model can be greatly reduced. The unknown parameters A, B, and W of the hybrid RBF network can be estimated by the recursive least square method [7]. Finally, this time-space synthesis can be used to reconstruct the spatiotemporal dynamics and predict the future outputs of the system.

References

[1] 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, Sep. 2010.

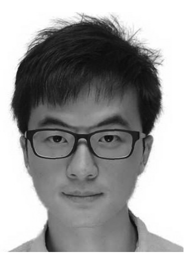
[2] J.-W. Wang and H.-N. Wu, "Exponential pointwise stabilization of semilinear parabolic distributed parameter systems via the Takagi–Sugeno fuzzy PDE model," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 1, pp. 155–173, Feb. 2018.

[3] J.-W. Wang, S.-H. Tsai, H.-X. Li, and H.-K. Lam, "Spatially piecewise fuzzy control design for sampled-data exponential stabilization of semilinear parabolic PDE systems," *IEEE Trans. Fuzzy Syst.*, to be published, doi: 10.1109/TFUZZ.2018.2809686.

[4] J.-W. Wang, H.-X. Li, and H.-N. Wu, "A membership-function-dependent approach to design fuzzy pointwise state feedback controller for nonlinear parabolic distributed parameter systems with spatially discrete actuators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1486–1499, Jul. 2017.

[5] 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, Jul. 2009.

[6] M. Wang, H.-X. Li, X. Chen, and Y. Chen, "Deep learning-based model reduction for distributed parameter systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 12, pp. 1664–1674, Dec. 2016.
[7] 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, Oct. 2009.
[8] Z. Liu and H.-X. Li, "A spatiotemporal estimation method for temperature distribution in lithium-ion batteries," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2300–2307, Nov. 2014.
[9] K.-K. Xu, H.-X. Li, and H.-D. Yang, "Kernel-based random vector functional-link network for fast learning of spatiotemporal dynamic processes," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2694018.
[10] K.-K. Xu, H.-X. Li, and Z. Liu, "ISOMAP-based spatiotemporal modeling for lithium-ion battery thermal process," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 569–577, Feb. 2018.
[11] 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. Syst.*, vol. 29, no. 2, pp. 457–469, Feb. 2018.
[12] D. Coca and S. A. Billings, "Identification of finite dimensional models of infinite dimensional dynamical systems," *Automatica*, vol. 38, no. 11, pp. 1851–1865, Nov. 2002.
[13] H. He, S. Chen, K. Li, and X. Xu, "Incremental learning from stream data," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1901–1914, Dec. 2011.
[14] G. A. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network," *Computer*, vol. 21, no. 3, pp. 77–88, Mar. 1988.
[15] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.
[16] Z. Wang, C. Chen, H.-X. Li, D. Dong, and T.-J. Tarn, "A novel incremental learning scheme for reinforcement learning in dynamic environments," in *Proc. 12th World Congr. Intell. Control Autom.*, Guilin, China, Jun. 2016, pp. 2426–2431, doi: 10.1109/WCICA.2016.7578530.
[17] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, May 2008.
[18] S. Yang and X. Yao, "Population-based incremental learning with associative memory for dynamic environments," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 542–561, Oct. 2008.
[19] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *Int. J. Robot. Res.*, vol. 31, no. 3, pp. 330–345, Mar. 2012.
[20] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive PCA for adaptive process monitoring," *J. Process Control*, vol. 10, no. 5, pp. 471–486, Oct. 2000.
[21] A. Varshney, S. Pitchaiah, and A. Armaou, "Feedback control of dissipative PDE systems using adaptive model reduction," *AIChE J.*, vol. 55, no. 4, pp. 906–918, Apr. 2009.
[22] D. B. Pourkargar and A. Armaou, "Modification to adaptive model reduction for regulation of distributed parameter systems with fast transients," *AIChE J.*, vol. 59, no. 12, pp. 4595–4611, Dec. 2013.
[23] D. B. Pourkargar and A. Armaou, "APOD-based control of linear distributed parameter systems under sensor/controller communication bandwidth limitations," *AIChE J.*, vol. 61, no. 2, pp. 434–447, Feb. 2015.
[24] C. Xu, L. Luo, and E. Schuster, "On recursive proper orthogonal decomposition via perturbation theory with applications to distributed sensing in cyber-physical systems," in *Proc. Amer. Control Conf.*, Baltimore, MD, USA, Jun./Jul. 2010, pp. 4905–4910, doi: 10.1109/ACC.2010.5530923.
[25] C. Xu and E. Schuster, "Model order reduction for high dimensional linear systems based on rank-1 incremental proper orthogonal decomposition," in *Proc. Amer. Control Conf.*, San Francisco, CA, USA, Jun./Jul. 2011, pp. 2975–2981, doi: 10.1109/ACC.2011.5991522.
[26] C. Xu and E. Schuster, "Low-dimensional modeling of linear heat transfer systems using incremental the proper orthogonal decomposition method," *Asia–Pac. J. Chem. Eng.*, vol. 8, no. 4, pp. 473–482, Jul./Aug. 2013.
[27] H.-X. Li and C. Qi, "Incremental modeling of nonlinear distributed parameter processes via spatiotemporal kernel series expansion," *Ind. Eng. Chem. Res.*, vol. 48, no. 6, pp. 3052–3058, Jan. 2009.
[28] L. Sirovich, *New Perspectives in Turbulence*, 1st ed. New York, NY, USA: Springer, 1991.
[29] J. Baker and P. D. Christofides, "Finite-dimensional approximation and control of non-linear parabolic PDE systems," *Int. J. Control*, vol. 73, no. 5, pp. 439–456, Nov. 2000.
[30] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems, and Symmetry*. New York, NY, USA: Cambridge Univ. Press, 1998.
[31] I. J. Leontaritis and S. A. Billings, "Input–output parametric models for non-linear systems part I: Deterministic non-linear systems," *Int. J. Control*, vol. 41, no. 2, pp. 303–328, 1985.
[32] J. Sjöberg *et al.*, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, Dec. 1995.
[33] X.-G. Zhou, L.-H. Liu, Y.-C. Dai, W.-K. Yuan, and J. L. Hudson, "Modeling of a fixed-bed reactor using the K-L expansion and neural networks," *Chem. Eng. Sci.*, vol. 51, no. 10, pp. 2179–2188, May 1996.
[34] N. Smaoui and S. Al-Enezi, "Modelling the dynamics of nonlinear partial differential equations using neural networks," *J. Comput. Appl. Math.*, vol. 170, no. 1, pp. 27–58, Sep. 2004.
[35] 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.
[36] E. Aggelogiannaki and H. Sarimveis, "Nonlinear model predictive control for distributed parameter systems using data driven artificial neural network models," *Comput. Chem. Eng.*, vol. 32, no. 6, pp. 1225–1237, Jun. 2008.
[37] H. Zha and H. D. Simon, "On updating problems in latent semantic indexing," *SIAM J. Sci. Comput.*, vol. 21, no. 2, pp. 782–791, 1999.
[38] A. Levy and M. Lindenbaum, "Sequential Karhunen–Loéve basis extraction and its application to images," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1371–1374, Aug. 2000.
[39] P. D. Christofides, *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport-Reaction Processes*. Boston, MA, USA: Birkhäuser, 2001.



Zhi Wang received the B.E. degree in automation from the Department of Control and Systems Engineering, Nanjing University, Nanjing, China, in 2015. He is currently pursuing the Ph.D. degree in machine learning and intelligent modeling with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong.

His current research interests include machine learning, deep learning, reinforcement learning, and system modeling.

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 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 a Professor with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong. He has a broad experience in both academia and industry. He has authored two books and seven patents, and published over 200 SCI journal papers with an H-index of 37 (Web of Science). His current research interests include process modeling and control, system intelligence, distributed parameter systems, and battery management systems.

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 SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS from 2002 to 2016, and IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS from 2009 to 2015. He serves as a Distinguished Expert for Hunan Government and the China Federation of Returned Overseas Chinese.