Incremental Learning Based Subspace Modeling for Distributed Parameter Systems

Zhi Wang
Department of Systems Engineering and Engineering Management, City University of Hong Kong
Hong Kong, China
njuwangzhi@gmail.com

Han-Xiong Li
Department of Systems Engineering and Engineering Management, City University of Hong Kong
Hong Kong, China
mehxli@cityu.edu.hk

***Abstract*—In this paper, a novel incremental learning based subspace modeling method is developed for spatiotemporal modeling of distributed parameter systems (DPSs). First, the streaming snapshots are collected into small batches at a preset time interval in an online mode. The initial batch belongs to the first nominal subspace. Second, the dissimilarity analysis is further utilized to assign each new batch to one of the existing subspaces or a new subspace. Third, the local basis functions corresponding to the assigned subspace is updated or generated through incremental learning of the new batch data. Finally, all the local models are ensembled to approximate the system's dynamics over the whole time-space domain in real-time. The proposed method is tested on a hyperbolic advection system and a one-dimensional diffusion-reaction system. Results demonstrate that the proposed method is superior to the conventional global modeling, and achieves higher modeling accuracy for DPSs.**

I. Introduction

MANY physical and chemical industrial processes (e.g. fluid flow, semiconductor manufacturing, curing thermal process and lithium-ion battery) belong to nonlinear distributed parameter systems (DPSs), which are described in partial differential equations (PDEs) with boundary and initial conditions [1]. These kinds of systems have strong spatiotemporal dynamics that are extremely difficult to model and control in system and control theory as well as industrial applications [2–4]. The time-space separation based methods [5–7], using spatial basis function expansion, have been verified to be an efficient model reduction method for unknown DPSs. In these spatiotemporal modeling methods, Karhunen-Loève decomposition (KLD) [8] 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.

However, traditional KLD is a linear dimension reduction method, it might not be very efficient for modeling nonlinear dynamics [9]. In addition, only one set of spatial BFs can be obtained from KLD. This fixed set of bases might not be very suitable for DPSs with complex nonlinear dynamics, because their dominant spatial bases might change. For this reason, the conventional spatiotemporal modeling method often leads to a poor nonlinear reduced-order model approximation [10]. An alternative to the global modeling approach is to capture nonlinear complexity by a combination of local linear KLD projections.

In this paper, we develop an integral incremental subspace modeling method based on dissimilarity analysis and local model updating for DPSs. The streaming snapshots are collected into small batches at a preset time interval in the online environment. Based on the idea that a change of operation condition can be detected by monitoring a distribution of process data, the dissimilarity analysis is utilized to assign the subspace that each new batch belongs to. If the new data batch is sufficiently different from all the existing subspaces, then a new subspace that the new batch belongs to is incrementally generated. Otherwise, if the new batch belongs to an existing subspace that derives a sufficiently small dissimilarity with the new batch, then the existing local model is updated through incremental learning of new data. The whole modeling structure is inherited and updated incrementally when the new data batch is available. At last, all the local models are ensembled to approximate the system's dynamics in real-time.

The advantages of our proposed incremental subspace modeling over conventional global modeling lie in three aspects:

1) By a combination of local linear KLD projections, the modeling framework can better approximate the nonlinear dynamics of many practical distributed processes.
2) Using incremental learning of process data in an online mode, the proposed method can track and adapt to the system dynamics in real-time.
3) Instead of repeatedly deriving BFs from scratch when new data arrives, the new method significantly improves computational efficiency by inheriting and updating the modeling structure in an incremental way.

In order to demonstrate the performances of the proposed method, simulation experiments are carried out on a hyperbolic advection system and a one-dimensional diffusion-reaction system. Both theoretical analysis and experimental results show that the proposed method achieves high modeling accuracy and efficiency, as well as being computationally effective.

II. Spatiotemporal Modeling

Modeling PDE systems can be synthesized into a time-space separation framework using KLD technique. For an easy

understanding, the following PDE

$$\frac{\partial y(x,t)}{\partial t} = \alpha \frac{\partial^2 y}{\partial x^2} + \beta \frac{\partial y}{\partial x} + f(y) + \boldsymbol{b}^T(x)\boldsymbol{u}(t), \quad (1)$$

with the boundary conditions $y(0,t) = 0, y(\pi,t) = 0$, and the initial condition $y(x,0) = y_0(x)$, will be selected as an example [1].

Consider a set of output data $\{y(x_i,t)\}_{i=1,t=1}^{n,l}$, where t is the time variable, $x \in \Omega$ is the spatial variable, and Ω is the spatial domain. The output is measured at the n spatial locations $x_1, ..., x_n$. Define the inner product, norm and ensemble average as $(f(x), g(x)) = \int_\Omega f(x)g(x)dx$, $||f(x)|| = (f(x), f(x))^{1/2}$ and $\langle f(x,t)\rangle = (1/l)\sum_{t=1}^{L} f(x,t)$.

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

$$y_k(x,t) = \sum_{i=1}^{k} \varphi_i(x) a_i(t), \quad (2)$$

where $y_k(x,t)$ denotes the kth-order approximation. The temporal coefficients can be computed from

$$a_i(t) = (\varphi_i(x), y(x,t)), i = 1, ..., k. \quad (3)$$

Finding the most characteristic spatial structure $\{\varphi_i(x)\}_{i=1}^k$ can be performed by minimizing the following Lagrangian function:

$$J = \langle ||y(x,t) - y_k(x,t)||^2 \rangle + \sum_{i=1}^{k} \lambda_i((\varphi_i, \varphi_i) - 1), \quad (4)$$

corresponding to constraints $(\varphi_i, \varphi_i) = 1, \varphi_i \in L^2(\Omega), i = 1, ..., k$. The necessary condition of the solution can be obtained as below

$$\int_\Omega R(x,\zeta)\varphi_i(\zeta)d\zeta = \lambda_i \varphi_i(x), (\varphi_i, \varphi_i) = 1, i = 1, ..., k, \quad (5)$$

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

A computationally efficient way to obtain the solutions of (5) is provided by the method of snapshots [11]. The eigenfunction $\varphi_i(x)$ is assumed to be expressed as a linear combination of the snapshots as follows:

$$\varphi_i(x) = \sum_{t=1}^{l} \gamma_{it} y(x,t). \quad (6)$$

Substituting (6) into (5) gives the following eigenvalue problem:

$$\int_\Omega \frac{1}{l} \sum_{t=1}^{l} y(x,t)y(\zeta,t) \sum_{k=1}^{l} \gamma_{ik} y(\zeta,k)d\zeta = \lambda_i \sum_{t=1}^{l} \gamma_{it} y(x,t). \quad (7)$$

Define the temporal two-point correlation function as

$$C_{tk} = \frac{1}{l} \int_\Omega y(\zeta,t)y(\zeta,k)d\zeta, \quad (8)$$

the eigenvalue problem (5) can be transformed to the following form of a $l \times l$ matrix eigenvalue problem:

$$\boldsymbol{C}\boldsymbol{\gamma}_i = \lambda_i \boldsymbol{\gamma}_i, \quad (9)$$

where $\boldsymbol{\gamma}_i = [\gamma_{i1}, ..., \gamma_{il}]^T$ is the ith eigenvector. The solution of the above eigenvalue problem yields the eigenvectors $\boldsymbol{\gamma}_1, ..., \boldsymbol{\gamma}_l$, which can be used in (6) to construct the eigenfunctions $\varphi_1(x), ..., \varphi_l(x)$. Since the matrix C is symmetric and positive semidefinite, the computed eigenfunctions are orthogonal.

The dominant k BFs $\{\varphi_i(x)\}_{i=1}^k$ is selected in the order of the magnitude of the corresponding eigenvalues, which can capture 99% of the system "energy" according to

$$E_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{l} \lambda_j}. \quad (10)$$

III. Incremental Subspace Modeling

A. Framework

As shown in Fig. 1, in the online environment, the streaming snapshots are collected into small batches $(..., D_{i-1}, D_i, ...)$ with the same window size (time interval T) in real-time. The initial batch of snapshots D_1 belongs to the first nominal subspace S_1, where D_1 is used to construct the corresponding local ROM M_1 by KLD. Assume that there are K existing subspaces. The dissimilarity indexes $Dis(D_i, S_j), (j = 1, 2, ...K)$ between the new batch D_i and all existing subspaces are computed to find out the Jth subspace that derives a minimal dissimilarity with the new batch. If $Dis(D_i, S_J) > \delta$, then the new batch is sufficiently different from subspace S_J and any other existing subspace. Consequently, a new subspace S_{K+1} should be generated that the new batch is assigned to. The corresponding local ROM M_{K+1} is constructed using the new data D_i by KLD. The generating of new subspaces forms the first part of the incremental learning mechanism. If $Dis(D_i, S_J) < \delta$, then the new batch D_i is confirmed to belong to subspace S_J. The existing local model M_J is further updated by incremental learning of new data D_i, which leads to the second part of the incremental learning mechanism.

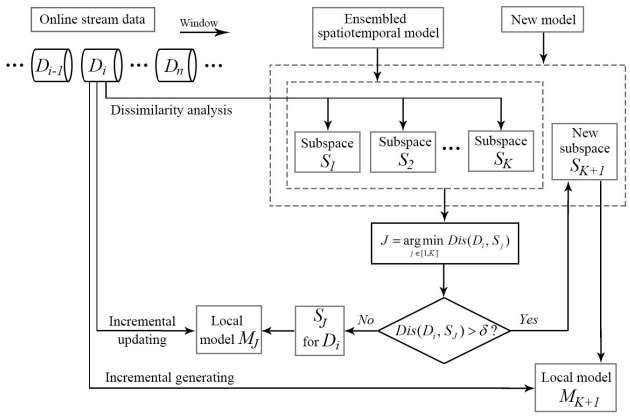


Fig. 1. Methodology framework of incremental subspace learning.

B. Dissimilarity Analysis

Dissimilarity analysis is a classification method based on KLD for evaluating the difference between distributions of data sets [12, 13]. Considering the following two data sets of snapshots: $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$, each column of which is an n-dimensional vector of spatial measurements from (1). $\boldsymbol{Y}_i, (i = 1, 2)$ consists of N_i columns of snapshots. The covariance matrices are given by $\boldsymbol{R}_i = \boldsymbol{Y}_i \boldsymbol{Y}_i^T / N_i$. The covariance matrix of the mixture of both data sets is given by

$$\boldsymbol{R} = \frac{N_1}{N_1 + N_2} \boldsymbol{R}_1 + \frac{N_2}{N_1 + N_2} \boldsymbol{R}_2. \tag{11}$$

On the basis of the fact that covariance matrix $\boldsymbol{R}$ can be diagonalized by an orthogonal matrix $\boldsymbol{P}_0$ as $\boldsymbol{P}_0^T \boldsymbol{R} \boldsymbol{P}_0 = \boldsymbol{\Lambda}$. The original data matrices $\boldsymbol{Y}_i$ are transformed into $\boldsymbol{Z}_i$

$$\boldsymbol{Z}_i = \sqrt{\frac{N_i}{N_1 + N_2}} \boldsymbol{\Lambda}^{-\frac{1}{2}} \boldsymbol{P}_0^T \boldsymbol{Y}_i = \sqrt{\frac{N_i}{N_1 + N_2}} \boldsymbol{P}^T \boldsymbol{Y}_i, \tag{12}$$

where $\boldsymbol{P}$ is a transformation matrix defined as $\boldsymbol{P} = \boldsymbol{P}_0 \boldsymbol{\Lambda}^{-\frac{1}{2}}$.

The covariance matrices of the transformed data matrices

$$\boldsymbol{S}_i = \frac{1}{N_i} \boldsymbol{Z}_i \boldsymbol{Z}_i^T = \frac{N_i}{N_1 + N_2} \boldsymbol{P}^T \boldsymbol{R}_i \boldsymbol{P}, \tag{13}$$

satisfy the equation $\boldsymbol{S}_1 + \boldsymbol{S}_2 = \boldsymbol{I}$. By application of eigenvalue decomposition to the covariance matrices

$$\boldsymbol{S}_i \boldsymbol{w}_i^j = \lambda_i^j \boldsymbol{w}_i^j, \quad 1 - \lambda_1^j = \lambda_2^j, \tag{14}$$

where λ_i^j and $\boldsymbol{w}_i^j$ are the eigenvalues and the corresponding eigenvectors, and the superscript j denotes the jth eigenvalue or eigenvector. The above two relationships mean the transformed data matrices, $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$, have the same set of principal components and the principal components are reversely ordered.

Finally, the index Dis is defined for evaluating the dissimilarity between snapshots $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$

$$Dis(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \frac{4}{n} \sum_{j=1}^{n} (\lambda_j - 0.5)^2, \tag{15}$$

where λ_j denotes the eigenvalues of the covariance matrix of the transformed data matrix. When data sets are quite similar to each other, the eigenvalues λ_j must be near 0.5, and then Dis should be near zero. On the other hand, when data sets are quite different from each other, the largest and the smallest eigenvalues should be near one and zero respectively. Hence, the index Dis changes between zero and one.

C. Incremental Local Model Updating

Assume that the subspace S_J contains l snapshots, denoted as $\boldsymbol{Y}_1 = [\boldsymbol{y}_1, ..., \boldsymbol{y}_l]$, where $\boldsymbol{y}_j, (j = 1, ..., l)$ is an n-dimensional snapshot of spatial measurements. By KLD, one can obtain kth-order dominant BFs $\{\varphi_i\}_{i=1}^k$ of this subspace. When the new data set D_i, containing m snapshots and denoted as $\boldsymbol{Y}_2 = [\boldsymbol{y}_{l+1}, ..., \boldsymbol{y}_{l+m}]$, belongs to subspace S_J, the corresponding local model M_J should be updated using D_i. For conventional modeling method, the BFs are retrained from scratch using the augmented data matrix $\boldsymbol{Y} = [\boldsymbol{Y}_1 \quad \boldsymbol{Y}_2]$. Calculating the Karhunen-Loève (KL) basis for $l + m$ snapshots of n spatial measurements requires roughly $O(n(l + m))$ memory units and $O((l + m)^3)$ flops. As the online process generates more and more historical data, the growing data length l results in superlinearly increasing computational complexity and linearly increasing storage capacity for the conventional method. Instead, an incremental learning approach based on the SVD-updating algorithm [14, 15] is adopted to update the local ROM efficiently.

By singular value decomposition (SVD), the matrix $\boldsymbol{Y}_1$ can be decomposed into $\boldsymbol{Y}_1 = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T$. According to (8), the temporal correlation matrix $\boldsymbol{C}$ can be rewritten as $\boldsymbol{C} = \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^T$, where $\boldsymbol{\Lambda} = \frac{1}{l} \boldsymbol{\Sigma} \boldsymbol{\Sigma}^T$ is a $l \times l$ diagonal matrix. According to (6), we can construct the k dominant BFs as $\boldsymbol{\Phi} = \boldsymbol{Y}_1 \boldsymbol{V}_k$, where $\boldsymbol{V}_k$ is formed by the first k columns of $\boldsymbol{V}$. Let the QR decomposition of $(\boldsymbol{I} - \boldsymbol{U}_k \boldsymbol{U}_k^T) \boldsymbol{Y}_2$ be

$$(\boldsymbol{I} - \boldsymbol{U}_k \boldsymbol{U}_k^T) \boldsymbol{Y}_2 = \boldsymbol{Q} \boldsymbol{R}, \tag{16}$$

where $\boldsymbol{Q}$ is orthonormal, $\boldsymbol{R}$ is the $m' \times m$ upper triangular, and m' ($m' \leq min(n, m)$) is the rank of $(\boldsymbol{I} - \boldsymbol{U}_k \boldsymbol{U}_k^T) \boldsymbol{Y}_2$. This step projects the new columns $\boldsymbol{Y}_2$ to the orthogonal complement of the old left latent space, i.e., span$\{\boldsymbol{U}_k\}$. It is verified that

$$\boldsymbol{Y} = [\boldsymbol{Y}_1 \quad \boldsymbol{Y}_2] = [\boldsymbol{U}_k \quad \boldsymbol{Q}] \begin{bmatrix} \boldsymbol{\Sigma}_k & \boldsymbol{U}_k^T \boldsymbol{Y}_2 \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_k^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_M \end{bmatrix}, \tag{17}$$

noticing that $[\boldsymbol{U}_k \quad \boldsymbol{Q}]$ is orthonormal. Now obtain the SVD of the $(k + m) \times (k + m')$ matrix

$$\begin{bmatrix} \boldsymbol{\Sigma}_k & \boldsymbol{U}_k^T \boldsymbol{Y}_2 \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} = [\widetilde{\boldsymbol{U}}_k \quad \widetilde{\boldsymbol{U}}_k^{\perp}] \begin{bmatrix} \bar{\boldsymbol{\Sigma}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{\Sigma}}_{M \times m'} \end{bmatrix} [\widetilde{\boldsymbol{V}}_k \quad \widetilde{\boldsymbol{V}}_k^{\perp}]^T, \tag{18}$$

where $\widetilde{\boldsymbol{U}}_k$ and $\widetilde{\boldsymbol{V}}_k$ are of column dimension k, and $\bar{\boldsymbol{\Sigma}}_k \in \mathbb{R}^{k \times k}$.

The best rank-k approximation of $\boldsymbol{Y}$, as $\boldsymbol{Y}_k$, is given by

$$\boldsymbol{Y}_k = ([\boldsymbol{U}_k \quad \boldsymbol{Q}] \widetilde{\boldsymbol{U}}_k) \bar{\boldsymbol{\Sigma}}_k (\begin{bmatrix} \boldsymbol{V}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_M \end{bmatrix} \widetilde{\boldsymbol{V}}_k)^T. \tag{19}$$

Then the best rank-k approximation of the new correlation matrix $\bar{\boldsymbol{C}}$ can be rewritten as

$$\begin{aligned} \bar{\boldsymbol{C}}_k &= \frac{1}{l + m} \boldsymbol{Y}_k^T \boldsymbol{Y}_k \\ &= \frac{1}{l + m} (\begin{bmatrix} \boldsymbol{V}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_M \end{bmatrix} \widetilde{\boldsymbol{V}}_k) \bar{\boldsymbol{\Sigma}}_k^2 (\begin{bmatrix} \boldsymbol{V}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_M \end{bmatrix} \widetilde{\boldsymbol{V}}_k)^T \\ &= \bar{\boldsymbol{V}}_k \bar{\boldsymbol{\Lambda}}_k \bar{\boldsymbol{V}}_k^T, \end{aligned} \tag{20}$$

where the updated diagonal matrix $\bar{\boldsymbol{\Lambda}}_k = \frac{1}{l+m} \bar{\boldsymbol{\Sigma}}_k^2$, and $\bar{\boldsymbol{V}}_k = (\begin{bmatrix} \boldsymbol{V}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_M \end{bmatrix} \widetilde{\boldsymbol{V}}_k)$. The k dominant BFs is updated as

$$\bar{\boldsymbol{\Phi}} = [\boldsymbol{Y}_1 \quad \boldsymbol{Y}_2] \bar{\boldsymbol{V}}_k = [\boldsymbol{\Phi} \boldsymbol{A} \quad \boldsymbol{Y}_2] (\begin{bmatrix} \boldsymbol{V}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_M \end{bmatrix} \widetilde{\boldsymbol{V}}_k), \tag{21}$$

where $\boldsymbol{A}$ is the acquired temporal coefficients matrix of $\boldsymbol{Y}_1$ using (3). In this incremental way, the old BFs $\boldsymbol{\Phi}$ is transformed to an updated one $\bar{\boldsymbol{\Phi}}$ when the new increment of snapshots $\boldsymbol{Y}_2$ arrives.

The two main computation steps are QR decomposition in (16) and SVD in (18), which take approximately $O(nm^2)$ and

$O((k+m')(k+m)^2)$ flops, respectively. In common cases, the number of dominant BFs k is usually much smaller than other parameters, i.e., $k \ll \{m', n, m\}$ and $m' \leq \min\{n, m\}$. The total time complexity of the incremental model updating procedure stays at the level of $O(nm^2)$, depending on the length m of data increment instead of the length l of historical data. Therefore, the incremental updating function (21) enables recursive calculation with significantly improved computational efficiency, which is important for online implementation [1, 16] of the proposed incremental subspace modeling.

IV. Simulation Experiments

To test the proposed method, two DPSs with different types of dynamics are studied: a hyperbolic advection system with strong nonlinearity, and a one-dimensional diffusion-reaction system with a time-varying domain. Two baselines are implemented: 1) the conventional global modeling that constructs one single ROM for all the snapshots; 2) "blind subspace modeling", which blindly partitions the snapshots into different subspaces and also construct a local ROM for each subspace. Define $y(x,t)$ and $y_k(x,t)$ as the measured output and the predicted output, respectively. The following performance indices are set up for modeling evaluation: 1)Spatiotemporal error $e(x,t) = y(x,t) - y_k(x,t)$; 2) Spatial normalized absolute error $\text{SNAE}(t) = \frac{1}{n}\sum_{i=1}^{n} |e(x_i,t)|$; 3) Root of mean squared error, $\text{RMSE} = (\int \sum e(x,t)^2 dx / \int dx \sum \Delta t)^{1/2}$.

A. Case 1: Hyperbolic Advection System

The advection is a common transport mechanism of a substance by energies, such as heat, humidity, or salinity. It can be formulated as a first-order hyperbolic PDE

$$\frac{\partial y}{\partial t} = -s\frac{\partial y}{\partial x}, \tag{22}$$

with the initial condition $y(x,0) = f(x)$, where s denotes the transportation speed. In this paper, a specific example with a smooth solution is considered by setting s to a constant and the initial condition to a Gaussian function

$$f(x) = e^{-\alpha(x+2s)^2}. \tag{23}$$

Then the traveling wave function of (22) can be obtained as:

$$y(x,t) = e^{-\alpha(x-s(t-2))^2}. \tag{24}$$

The parameters are set as $s=3$, $\alpha=10$. The measured snapshots $y(x,t)$ sampled at time interval $\Delta t = 0.001$ are shown in Fig. 2, from eighteen uniformly distributed sensors along the space domain $x \in [-2, 2]$. The stream snapshots are collected every 0.5 seconds in real-time, and every small batch $D_i, (i = 1, 2, ...)$ consists of 500 snapshots. The dissimilarity threshold σ is set as 0.5 for this case, which creates 4 subspaces as $(D_1, D_2, D_3) \in S_1$, $D_4 \in S_2, D_5 \in S_3, (D_6, D_7, D_8) \in S_4$.

First, the above three methods are applied to derive the low-order model of the snapshots with five dominant BFs selected. With the spatial BFs, the space-time synthesis can be conducted to obtain the spatiotemporal distribution which

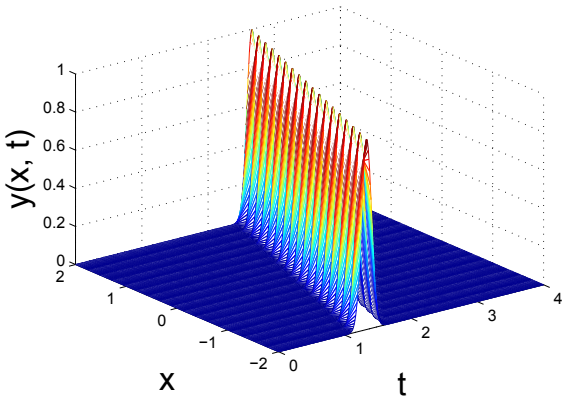


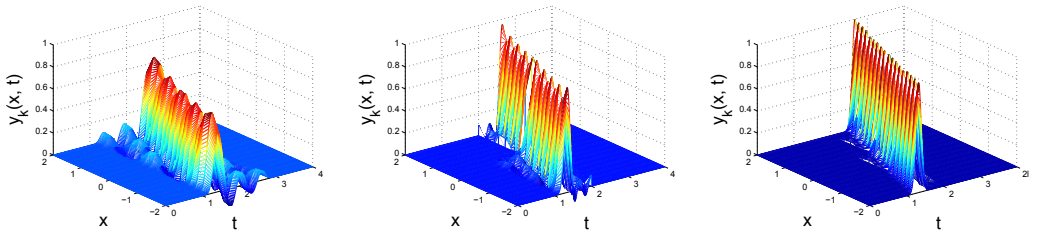
Fig. 2. Case 1: Measured output in time domain $t \in (0, 4]$.

is shown in Fig. 3. Correspondingly, the spatiotemporal prediction error and the spatial normalized absolute error over the whole time-space domain are illustrated in Figs. 4 and 5, respectively. It can be observed that the conventional global modeling performs poorly since the system's output is seriously distorted. The strong nonlinearity of such hyperbolic system makes the linear global model reduction method ineffective in capturing the intrinsic local characteristics. Compared to the global method, the blind subspace modeling gains an improved modeling accuracy by taking advantage of several sets of BFs. However, the system's dynamics are still not reconstructed accurately since the spatiotemporal error and the spatial normalized absolute error are still relatively high. Such unsatisfactory performance is caused by the limitation that snapshots with largely different characteristics may be assigned to the same subspace. In contrast, the proposed incremental subspace modeling shows high reconstruction accuracy with rather smaller approximation errors. The reduced-order solution computed via the proposed method is much closer to the original full-order solution. The improved performance benefits from the dissimilarity analysis that assigns snapshots with a large difference to different subspaces.

The modeling efficiency is studied and presented in Fig. 6. The proposed method can achieve higher accuracy with fewer number of modes, while the other two methods perform poorly since their accuracies increase slowly as the modes number increases. The presented RMSE profiles clearly show that the ROM constructed using 3 BFs from the proposed method provides a better approximation to the original system than the ROM derived using 8 BFs from the global method or 5 BFs from the blind subspace method. The proposed method provides a decent reduction in the number of BFs required to accurately represent the full-order solution, and demonstrates high model reduction efficiency in this paper.

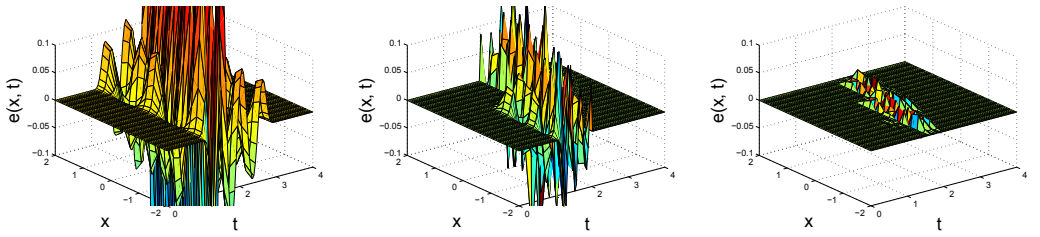
B. Case 2: 1-D Diffusion-reaction System

In this section, we present an application of the proposed method to a diffusion-reaction process with nonlinearities, spatially-varying coefficients and a time-dependent spatial



(a) Global modeling (b) Blind subspace modeling (c) Proposed incremental subspace modeling

Fig. 3. Case 1: Spatiotemporal prediction by the three methods.



(a) Global modeling (b) Blind subspace modeling (c) Proposed incremental subspace modeling

Fig. 4. Case 1: Spatiotemporal error by the three methods.

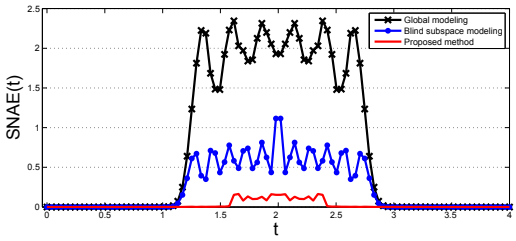


Fig. 5. Case 1: Spatial normalized absolute error by the three methods.

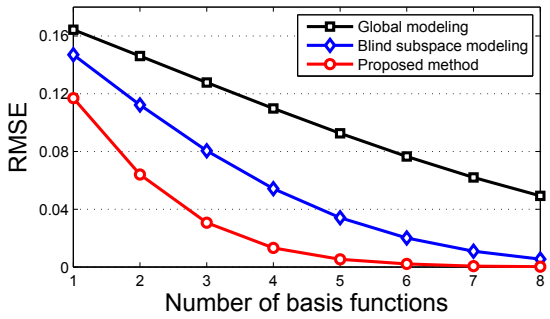


Fig. 6. Case 1: Modeling accuracy comparison under a different number of modes.

domain. Specifically, we consider a diffusion-reaction process which is described by the following parabolic PDE

$$\frac{\partial y}{\partial t} = \frac{\partial}{\partial x}(k(x)\frac{\partial y}{\partial x}) + \beta_T(x)(e^{-\frac{\gamma}{1+y}-e^{-\gamma}}) + \beta_U(\boldsymbol{b}^T(x)\boldsymbol{u}(t) - y), \tag{25}$$

subject to the Dirichlet boundary conditions: $y(0,t) = 0, y(l(t),t) = 0, y(x,0) = 0.5$. The values and expressions of the process parameters are set as $\beta_U = 2, \gamma = 4, \beta_T(x) = 45(1.5\text{–}e^{-0.5x})$, $k(x) = e^{-0.5x}$ and $l(t) = \pi[1.4\text{–}0.4e^{-0.02t^{2.7}}]$. There are available four actuators $\boldsymbol{u}(t) = [u_1(t), ..., u_4(t)]^T$ with the spatial distribution function $\boldsymbol{b}(x) = [b_1(x), ..., b_4(x)]^T$, $b_i(x) = H(x-(i-1)\pi/4) - H(x-i\pi/4)$, $(i = 1, ..., 4)$ and $H(\cdot)$ is the standard Heaviside function. The temporal input is set as a series of sinusoidal signals with different frequencies $u_i(t) = 1.1 + 5sin(t + i/10)$.

In this case, the snapshots are sampled at time interval $\Delta t = 0.01$ along the time-varying spatial domain. The measured output for $t \in (0, 40]$ is shown in Fig. 7. The streaming snapshots are collected every 2 seconds in real-time, and every small batch $D_i, (i = 1, 2, ...)$ consists of 200 snapshots. The dissimilarity threshold σ is set as 0.5 for this case, which creates 4 subspaces as $D_1 \in S_1$, $D_2 \in S_2, D_3 \in S_3, (D_4, D_5, ..., D_{20}) \in S_4$. Intuitively, the incremental subspace modeling assigns the snapshots to different subspaces that correspond to the time-varying spatial domain.

Three dominant BFs are selected for the tested methods to compare their reconstruction accuracy. The spatial normalized absolute error over the whole time/space domain are illustrated in Fig. 8. It can be observed that the proposed method greatly improves modeling accuracy by providing much smaller approximation errors than the other two methods.

V. Conclusions

In the paper, an integral incremental subspace modeling approach based on dissimilarity analysis and local model updating is presented for DPSs. The snapshots collected in

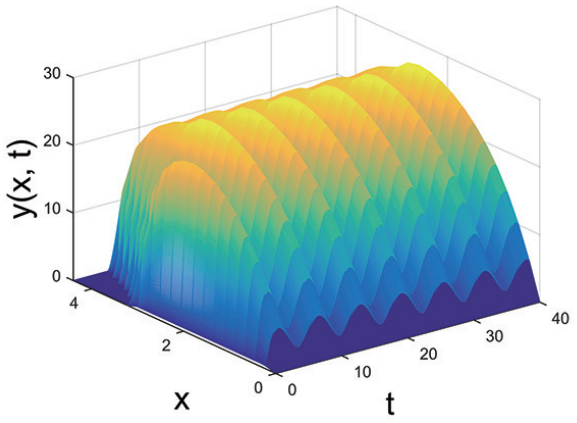


Fig. 7. Case 2: Measured snapshots with the time-varying domain.

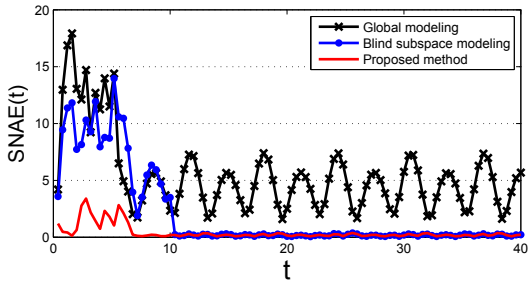


Fig. 8. Case 2: Spatial normalized absolute error by the three methods.

real-time are assigned to different subspaces by dissimilarity analysis. Then the local ROM corresponding to the assigned subspace is updated or generated through incremental learning of the new data. Finally, all the local models are ensembled to predict the system's dynamics over the whole time-space domain. By a combination of local projections in an online model, the proposed method can track and adapt to the system dynamics in real-time. The proposed method is successfully applied to a hyperbolic advection and a one-dimensional diffusion-reaction system. Simulation results demonstrate that the proposed method achieves much high modeling accuracy and efficiency. Future implementation of better subspace partitioning methods is under consideration.

REFERENCES

[1] Z. Wang and H.-X. Li, "Incremental spatiotemporal learning for online modeling of distributed parameter systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018. doi: 10.1109/TSMC.2018.2810447.

[2] Y. Feng and H.-X. Li, "Detection and spatial identification of fault for parabolic distributed parameter systems," *IEEE Transactions on Industrial Electronics*, 2018. doi: 10.1109/TIE.2018.2877188.

[3] Z. Wang, L. Han-Xiong, and C. Chen, "Reinforcement learning based optimal sensor placement for spatiotemporal modeling," *IEEE Transactions on Cybernetics*, 2019. doi: 10.1109/TCYB.2019.2901897.

[4] Y. Feng and H.-X. Li, "Dynamic spatial independent component analysis based abnormality localization for distributed parameter systems," *IEEE Transactions on Industrial Informatics*, 2019. doi: 10.1109/TII.2019.2900226.

[5] X.-B. Meng, H.-X. Li, and H.-D. Yang, "Evolutionary design of spatiotemporal learning model for thermal distribution in lithium-ion batteries," *IEEE Transactions on Industrial Informatics*, 2018. doi: 10.1109/TII.2018.2866468.

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

[7] B.-C. Wang and H.-X. Li, "A sliding window based dynamic spatiotemporal modeling for distributed parameter systems with time-dependent boundary conditions," *IEEE Transactions on Industrial Informatics*, 2018. doi: 10.1109/TII.2018.2859444.

[8] D. Zheng, K. A. Hoo, and M. J. Piovoso, "Low-order model identification of distributed parameter systems by a combination of singular value decomposition and the karhunen- loève expansion," *Industrial & Engineering Chemistry Research*, vol. 41, no. 6, pp. 1545–1556, 2002.

[9] E. C. Malthouse, "Limitations of nonlinear PCA as performed with generic neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 165–173, 1998.

[10] D. Amsallem, M. J. Zahr, and C. Farhat, "Nonlinear model order reduction based on local reduced-order bases," *International Journal for Numerical Methods in Engineering*, vol. 92, no. 10, pp. 891–916, 2012.

[11] L. Sirovich, *New perspectives in turbulence (1st ed.)*. New York: Springer, 1991.

[12] M. Kano, S. Hasebe, I. Hashimoto, and H. Ohno, "Statistical process monitoring based on dissimilarity of process data," *AIChE Journal*, vol. 48, no. 6, pp. 1231–1240, 2002.

[13] C. Zhao, F. Wang, and M. Jia, "Dissimilarity analysis based batch process monitoring using moving windows," *AIChE journal*, vol. 53, no. 5, pp. 1267–1277, 2007.

[14] H. Zha and H. D. Simon, "On updating problems in latent semantic indexing," *SIAM Journal on Scientific Computing*, vol. 21, no. 2, pp. 782–791, 1999.

[15] A. Levey and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," *IEEE Transactions on Image processing*, vol. 9, no. 8, pp. 1371–1374, 2000.

[16] Z. Wang, C. Chen, H.-X. Li, D. Dong, and T.-J. Tarn, "Incremental reinforcement learning with prioritized sweeping for dynamic environments," *IEEE/ASME Transactions on Mechatronics*, 2019. doi: 10.1109/TMECH.2019.2899365.