Online Identification of Nonlinear Spatiotemporal Systems Using Kernel Learning Approach

Hanwen Ning, Xingjian Jing, and Li Cheng

Abstract— **The identification of nonlinear spatiotemporal systems is of significance to engineering practice, since it can always provide useful insight into the underlying nonlinear mechanism and physical characteristics under study. In this paper, nonlinear spatiotemporal system models are transformed into a class of multi-input–multi-output (MIMO) partially linear systems (PLSs), and an effective online identification algorithm is therefore proposed by using a pruning error minimization principle and least square support vector machines. It is shown that many benchmark physical and engineering systems can be transformed into MIMO-PLSs which keep some important physical spatiotemporal relationships and are very helpful in the identification and analysis of the underlying system. Compared with several existing methods, the advantages of the proposed method are that it can make full use of some prior structural information about system physical models, can realize online estimation of the system dynamics, and achieve accurate characterization of some important nonlinear physical characteristics of the system. This would provide an important basis for state estimation, control, optimal analysis, and design of nonlinear distributed parameter systems. The proposed algorithm can also be applied to identification problems of stochastic spatiotemporal dynamical systems. Numeral examples and comparisons are given to demonstrate our results.**

Index Terms— **Lattice dynamics, least squares support vector machines, nonlinear system identification, partially linear systems, spatiotemporal systems.**

I. Introduction

SPATIO-TEMPORAL systems given by partial differential equations (PDEs) are widely used to describe physical and engineering phenomena such as heat process, population dynamics, chemical reactors, fluid dynamics, etc., [1], [2]. Traditional methods for the analysis of a PDE system relies on an analytical solution of the system, which is actually difficult to obtain for most nonlinear PDEs. Alternatively, qualitative solutions such as the existence, stability, controllability, observability etc., are studied by using functional analysis, Sobolev space theory, generalized function theory, etc., [3]–[7]. Numerical methods are also employed to find an approximation of the solution of a PDE system, which include finite element methods and difference methods. Noticeably, the difference method is to translate PDEs into lattice dynamical systems (or multidimensional difference equations) [8]–[10]. The concept of lattice dynamical systems adopts state space expressions and has been applied to various systems, and is found to be powerful to reproduce complex spatiotemporal patterns [11], [12]. The variables with respect to each node on the lattice represent the same set of physical quantities. If the numeral relationship of these variables could be obtained, then the values of the dependent variables and the approximate solution of the PDE could be derived iteratively. These benefits in solving numerical solution, predicting system future states or system states that are not available for measurement and maintaining a straightforward link to the physical properties of the original system, provide a basic motivation of the technical method developed in this paper.

Note that identification of spatiotemporal dynamical systems has also been studied recently by using finite-dimensional parametric multi-input–multi-output (MIMO) models such as nonlinear autoregressive exogenous model (NARX) and neural networks to approximate infinite-dimensional systems [13]–[17]. In these results, the estimation of spatiotemporal systems is technically formulated into a traditional identification problem of a MIMO system, and therefore many existing methods in nonlinear system identification theory can be applied, such as set membership, orthogonal least squares, etc., [18]–[21]. Because multiple input and output variables are involved in regressors including their nonlinear combinations, existing methods are usually computationally intensive and difficult (if not impossible) to be applied for online estimation. However, it is observed that with some known spatiotemporal structural information about systems, the PDE model can eventually be transformed into a partially linear model with a linear part of known structure and an unknown nonlinear part. For the transformed partially linear model, the known model structure could be a useful factor to reduce algorithm complexity, and thus online estimation could be achieved by using kernel learning approaches to approximate the nonlinear part. Among the kernel learning methods [22]–[24], the support vector machine (SVM) method is a promising statistical learning theory first developed by Vapnik [25], which can realize the best estimation with the least number of samples [26]–[28], and has relatively good performance when used for MIMO systems [29]. These shed light on some other motivations of the technical method to be developed in this paper.

Moreover, this paper also aims at methods that can accurately estimate system physical characteristics which are

Manuscript received November 17, 2010; accepted June 27, 2011. Date of publication July 22, 2011; date of current version August 31, 2011. This work was supported in part by the General Research Fund Project of Hong Kong RGC under Ref. 517810, Department of General Research Funds and Competitive Research Grants, Hong Kong Polytechnic University.

The authors are with the Department of Mechanical Engineering, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: mmjxj@polyu.edu.hk; xingjian.jing@gmail.com).

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/TNN.2011.2161331

nonlinear spatial-dependent functions characterized by some model parameters in the PDE system. This will provide an important insight into the analysis and design of physical and structural properties of the dynamical system under study.

Therefore, based on pruning minimization principles [30] and the least squares SVM (LS-SVM), an effective identification algorithm named pruning minimization principle recursive least squares-support vector machine PM-RLS-SVM is proposed for the online identification of spatiotemporal systems from the perspective of MIMO partially linear models (PLMs). It is shown that benchmark spatiotemporal systems in heat process and vibration systems can always be transformed into lattice dynamic systems and then into MIMO PLMs, which have a clear structural information of the data, and thus allow accurate estimations of the model and important model parameters (which have direct links with physical, material, or structural properties) simultaneously. In contrast to several existing identification methods for spatiotemporal systems [13]–[17], the computation complexity is reduced by using the prior spatiotemporal structural information of the data and the powerful LS-SVM method. Although the SVM methods have already been applied in solving different problems, to the best of the authors' knowledge, few results are available for online identification problems of nonlinear spatiotemporal systems. Comparisons with several existing and recently developed kernel learning methods are conducted to demonstrate the advantages of the new learning algorithm. Necessary discussions are also provided to illustrate the effectiveness and to point out potential applications of our results.

This paper is organized as follows. Lattice dynamical systems are introduced in Section II with illustrations for benchmark spatiotemporal systems. Section III introduces the MIMO partially linear model. In Section IV, based on LS-SVM and pruning error minimization principle, an online algorithm for MIMO partially linear model is proposed. Section V provides numerical examples and comparisons to demonstrate the new method. Conclusions are provided at the end.

II. Discretization and Lattice Dynamic Systems

Exact solutions for most of the nonlinear PDEs are difficult to obtain due to diverse nonlinearity, different structures, and complex boundary conditions. To estimate state values at any time and space positions, numeral methods (e.g., classic difference methods) can be adopted. The discrete lattice difference model has good performance in simulating spatiotemporal dynamic systems [11], [12], [31]. The basic idea is to replace the differential quotient by the difference quotient, and finally to construct a proper lattice dynamic structure for the specific spatiotemporal system to find approximate values using iteration methods [8]–[10].

To demonstrate this, consider a heat transfer system

$$\begin{cases} \frac{\partial u}{\partial t} = a(x)\frac{\partial^2 u}{\partial x^2} + f(u), \quad 0 < t \le T, \\ u(x,0) = \xi(x), \quad 0 < x < 1, \\ u(0,t) = u(1,t) = 0, \quad 0 \le t \le T \end{cases} \tag{1}$$

where f is a unknown nonlinear function. Let $0 = x_0 < x_1 < \cdots < x_n = 1$, $0 = t_0 < t_1 < \cdots < t_m = T$, $h = x_{j+1} - x_j$, $j = 0, 1, \ldots, n-1$, $\tau = t_{k+1} - t_k$, and $u(x_j, t_k) = u_j^k$. With this discretization method, we can obtain

$$\begin{aligned} &\frac{u(x_j, t_{k+1}) - u(x_j, t_k)}{\tau} \\ &\quad \approx \left.\frac{\partial u}{\partial t}\right|_{(x_j,t_k)} = a(x)\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_j,t_k)} + f(u(x_j,t_k)) \\ &\quad \approx a(x_j)\frac{u(x_{j+1},t_k) - 2u(x_j,t_k) + u(x_{j-1},t_k)}{h^2} \\ &\qquad + f(u(x_j,t_k)) \end{aligned} \tag{2}$$

which further yields

$$u_j^{k+1} = r_j u_{j-1}^k + (1-2r_j)u_j^k + r_j u_{j+1}^k + \tau f(u_j^k) \tag{3}$$

where $r_j = a_j\frac{\tau}{h^2}$ (denote $a(x_j) = a_j$). We can specify it with the matrix form

$$\begin{bmatrix} u_1^{k+1} \\ u_2^{k+1} \\ . \\ . \\ u_{n-1}^{k+1} \end{bmatrix} = \tau \begin{bmatrix} f(u_1^k) \\ f(u_2^k) \\ . \\ . \\ f(u_{n-1}^k) \end{bmatrix} + \begin{bmatrix} 1-2r_1 & r_1 & 0 & & 0 \\ r_2 & 1-2r_2 & r_2 & & 0 \\ & . & . & & \\ & & . & . & \\ 0 & 0 & . & r_{n-1} & 1-2r_{n-1} \end{bmatrix} \begin{bmatrix} u_1^k \\ u_2^k \\ . \\ . \\ u_{n-1}^k . \end{bmatrix}. \tag{4}$$

It is observed that this is a state space model with a combination of a linear part and a nonlinear part, where the parameters in the linear part, i.e., $r_j = a_j(\tau/h^2)$, have a direct link with the parameter $a(x)$ in the original physical model.

Similarly, consider a vibration system of two dimensions

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} = b(x,y)\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + g(u), \quad 0 < t \le T, \\ u(x,0,t) = \xi_1(x,t), \quad u(x,1,t) = \xi_2(x,t), \\ u(0,y,t) = \eta_1(t), \quad u(1,y,t) = \eta_2(y,t), \\ u(x,y,0) = \xi(x,y), \\ \frac{\partial u}{\partial t}(x,y,0) = \eta(x,y), \quad 0 < t < T, \ \ 0 \le x, y \le 1. \end{cases} \tag{5}$$

Let $0 = x_0 < x_1 < \cdots < x_{n_1} = 1$, $0 = y_0 < y_1 < \cdots < y_{n_2} = 1$, $0 = t_0 < t_1 < \cdots < t_m = T$, $h_1 = x_{j+1} - x_j$, $j = 0, 1, \ldots, n_1 - 1$, $h_2 = y_{j+1} - y_j$, $j = 0, 1, \ldots, n_2 - 1$, $\tau = t_{k+1} - t_k$, and $u(x_i, y_j, t_k) = u_{i,j}^k$. With this discretization method, we can obtain

$$\begin{aligned} u_{i,j}^{k+1} &= r_{i,j}^x(u_{i,j+1}^k + u_{i,j-1}^k) + 2(1 - r_{i,j}^x - r_{i,j}^y)u_{i,j}^k \\ &\quad + r_{i,j}^y(u_{i-1,j}^k + u_{i+1,j}^k) - u_{i,j}^{k-1} + \tau^2 g(u_{i,j}^k) \end{aligned} \tag{6}$$

where $r_{i,j}^x = b(x_i, y_j)(\tau^2/h_1^2)$ and $r_{i,j}^y = b(x_i, y_j)(\tau^2/h_2^2)$, which directly come from the model parameters of the original system. Since $u_{i,j}^{-1}$ is not given, $u_{i,j}^1$ cannot be computed. We approximate the initial condition on the derivative by $(u_{i,j}^1 - u_{i,j}^{-1})/\tau = \eta(x_i, y_j)$. Thus the term $u_{i,j}^{-1}$ can be removed. Actually, it is not necessary to know the data at the beginning or the boundary in the identification process.

It can be seen that the spatiotemporal dynamic systems (1) and (5) can both be translated into lattice dynamic systems using the difference approach, and the state space equations,

consisting of a linear part and a nonlinear part, clearly show the relationship of each node in the lattice with direct links to system parameters of original physical models. For these systems in practice, the nonlinear mappings f and g and parameters $a(x)$ and $b(x, y)$, which have particular physical meanings, could be completely unknown, and only some measurement data for u_j^k can be available. System identification is to construct a model to fit the data and predict future state of u at each lattice site. Since the data samples could be described by some general lattice structures mentioned above, this prior information can be utilized in system identification, and thus is helpful to build a more appropriate model. On the other hand, if the linear part of the lattice models can be obtained, the nonlinear functions [$a(x)$ and $b(x, y)$], which are linked to the physical properties of the systems, can therefore be identified. This will be advantageous and beneficial in practice compared with some black-box modeling methods.

III. MIMO Partially Linear Systems

Model structure selection is an important topic in nonlinear system identification [31], [32]. If some of the regressors are linear and some are nonlinear, it could be conducted in an alternative way (via partially linear models): that is, to exploit the known information on model structure as demonstrated before. Empirically, the complexity and generalizations in system identification could be improved by using a partially linear model structure with the same samples [33].

To this aim, spatiotemporal models will be formulated into a general partially linear model via the lattice dynamic system using state space equations. Consider a MIMO system

$$Y(k) = F[Y(k-1), \ldots, Y(k-n_y), U(k-1), \ldots, U(k-n_u)] + E(k) \tag{7}$$

where F is an unknown nonlinear mapping. $Y(k)$, $U(k)$, and $E(k)$ are the system output, input, and noise vector at time k. n_y and n_u are the maximal time lags. For (7), denote

$$Y(k) = [y_1(k), \ldots, y_M(k)]^T, U(k) = [u_1(k), \ldots, u_L(k)]^T, \\ E(k) = [e_1(k), \ldots, e_M(k)]^T \tag{8}$$

where $y_m(k)$ and $e_m(k)$ represent the mth output and noise of the system, respectively. $u_l(k)$ represents the lth input of the system. M is the number of output channels, and $m = 1, \ldots, M$. L is the number of inputs, and $l = 1, \ldots, L$. Thus, a general input vector consisting of all the possible variables existing in the regression could be constructed as follows:

$$x(k) = \left[Y^T(k-1), \ldots, Y^T(k-n_y), U^T(k-1) \ldots, U^T(k-n_u)\right]^T. \tag{9}$$

For each channel

$$y_m(k) = F_m(x(k)) + e_m(k) \tag{10}$$

for $m = 1, \ldots, M$. This leads to a MIMO NARX regression model, which includes all the possible combinations of linear and nonlinear terms, and it usually involves a complicated and time-consuming process to select the best regressors in system identification. It is evident from Section II that some of the regressors in the lattice dynamic models are linear. Therefore, it is reasonable to formulate into a partially linear model.

Let $X = \{x : x \text{ is a component of the vector } x(k)\}$. For any given channel m, define an arbitrary partition $X = X^{m_1} \bigcup X^{m_2}$ with $X^{m_1} \bigcap X^{m_2} = \emptyset$. The subscript m_1 and m_2 respectively represent the subset of regressors that linearly and nonlinearly enters into the regression model. $X^{m_1} \bigcap X^{m_2} = \emptyset$ is required to guarantee the uniqueness of the partition. For example, in the system $y_1(k) = y_1(k-1)^2 + y_1(k-1) + y_2(k-1)$, $y_1(k-1)$ must be classified as a nonlinear regressor and the system can be identified as $y_1(k) = y_2(k-1) + f(y_1(k-1))$. Thus, the regressor vector can be partitioned as $x(k) = \{(x^{m_1}(k))^T, (x^{m_2}(k))^T\}^T$. The MIMO partially linear model can be expressed as

$$y_m(k) = \beta_m^T x^{m_1}(k) + f_m(x^{m_2}(k)) \tag{11}$$

for $m = 1, \ldots, M$. Here, f_m's are nonlinear mappings. From the discretization formulas in the last section, the spatiotemporal lattice dynamical systems could be considered as a special case of MIMO partially linear models as long as the nodes of the lattice are marked properly. Note that the parameters in (1) and (5), which have physical meanings, are represented by the coefficients in the linear part.

IV. PM-RLS-SVM for MIMO Partially Linear Systems

A general identification algorithm for the MIMO partially linear system (11) is developed in this section. Consider a given set of training samples $\{x(k), Y(k)\}_{1,\ldots,N}$, i.e., $\{x^{m_1}(k), x^{m_2}(k), Y(k)\}_{1,\ldots,N}$, for the partially linear system, and the task is to find the underlying relationship of the dataset characterized by β_m and $f_m(.)$. For an SVM and the mth channel, the following regression model can be obtained:

$$y_m(k) = \beta_{m^T x}^{m_1}(k) + W_m^T \varphi_m(x^{m_2}(k)) + c_m. \tag{12}$$

Here, $\beta_m, x^{m_1}(k) \in R^{N_{m_1}}$, $x^{m_2}(k) \in R^{N_{m_2}}$, N_{m_1}, and N_{m_2} denote the number of regressors of the linear and nonlinear parts of the mth channel, respectively, and c_m is the bias term. The nonlinear mapping φ maps R^{m_2} into spaces R^{m_h}, which may be infinite dimensional and is assumed to satisfy the Mercer kernel condition [25]. The approximation error is defined as

$$e_m(k) = y_m(k) - \overline{y_m(k)} \tag{13}$$

where $\overline{y_m(k)}$ denotes the prediction for $y_m(k)$. The LS-SVM is used to find the weights that give the smallest summed quadratic error of the training samples, and a regularization strategy (ridge regression) is also needed to smoothen the approximation. To this aim, the following constrained optimization problem is constructed (prime problem):

$$J_m = \min_{W_m, c_m, e_m(k), \beta_m} \frac{1}{2} W_m^T W_m + \frac{1}{2\gamma} \sum_{k=1}^{N} e_m(k)^2 \tag{14}$$

with equality constraints

$$y_m(k) = \beta_m^T x^{m_1}(k) + W_m^T \varphi_m(x^{m_2}(k)) + c_m + e_m(k) \tag{15}$$

for $k = 1, \ldots, N$. The relative importance between the smoothness of the solution and the data fitting is determined by γ, which is a positive regularization constant. To solve the constrained optimization problem (14) and (15), a Lagrangian is formulated

$$\mathbf{L}_m = \frac{1}{2} W_m^T W_m + \frac{1}{2\gamma} \sum_{k=1}^{N} e_m(k)^2 - \sum_{k=1}^{N} \alpha_m(k)(\beta_m^T x^{m_1}(k) + W_m^T \varphi_m(x^{m_2}(k)) + c_m + e_m(k) - y_m(k)) \quad (16)$$

where $\alpha_m(k)$'s are the Lagrangian multipliers. To find the saddle point, the following hold:

$$\begin{aligned}
\frac{\partial \mathbf{L}_m}{\partial W_m} = 0 \quad &\Rightarrow \quad W_m = \sum_{k=1}^{N} \alpha_m(k) \varphi_m(x^{m_2}(k)) \\
\frac{\partial \mathbf{L}_m}{\partial c_m} = 0 \quad &\Rightarrow \quad \sum_{k=1}^{N} \alpha_m(k) = 0 \\
\frac{\partial \mathbf{L}_m}{\partial e_m(k)} = 0 \quad &\Rightarrow \quad \alpha_m(k) = \frac{1}{\gamma} e_m(k), \quad \text{for } k = 1, \ldots, N \\
\frac{\partial \mathbf{L}_m}{\partial \beta_m} = 0 \quad &\Rightarrow \quad \sum_{k=1}^{N} \alpha_m(k) x^{m_1}(k) = 0 \\
\frac{\partial \mathbf{L}_m}{\partial \alpha_m(k)} = 0 \quad &\Rightarrow \quad \text{for } k = 1, \ldots, N, \\
y_m(k) &= \beta_m^T x^{m_1}(k) + W_m^T \varphi_m(x^{m_2}(k)) + c_m + e_m(k). \quad (17)
\end{aligned}$$

By Mercer's theorem [25], $\varphi_m(x^{m_2}(k))^T \varphi_m(x^{m_2}(k)) = K_m(x^{m_2}(k), x^{m_2}(k))$ with a positive definite kernel K_m. After elimination of W_m and $e_m(k)$, we obtain

$$y_m(k) = \beta_m^T x^{m_1}(k) + \sum_{j=1}^{N} \alpha_m(k) K_m(x^{m_2}(j), x^{m_2}(k)) + c_m + \gamma\, \alpha_m(k). \quad (18)$$

If constructing a kernel matrix Ω_m with $\Omega_m(i, j) = K_m(x^{m_2}(i), x^{m_2}(j))$ $i, j = 1, \ldots, N$, then the coefficients of the regression model can be derived by solving the following dual problem:

$$\begin{bmatrix} 0 & 0 & X_{m_1}^T \\ 0 & 0 & 1^T \\ X_{m_1} & 1 & \Omega_m + \gamma I \end{bmatrix} \cdot \begin{bmatrix} \beta_m \\ c_m \\ \alpha_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ y_m \end{bmatrix} \quad (19)$$

where $X_{m_1} = [x^{m_1}(1)^T; x^{m_1}(2)^T; \ldots; x^{m_1}(N)^T] \in N \times R^{N_{m_1}}$ and $y_m = [y_m(1), y_m(2), \ldots, y_m(N)]$. For convenience, denote the matrix above (in the left side of the equation) consisting of N samples as $A_{m,N}$. Positive kernels must be chosen to guarantee the existence of a solution for the linear matrix equation. Obviously, the nonlinear mapping φ need not be defined explicitly. The most commonly used kernels satisfying Mercer's theorem are Gaussian radial basis functions (RBFs), polynomials, splines, etc., [25], [34]. The method described above could be applied to every output channel (11). Consequently, the estimated model for the MIMO partially linear system is given by

$$\begin{aligned}
\hat{y}_m(k) &= \beta_m^T x^{m_1}(k) + W_m^T \varphi_m(x^{m_2}(k)) + c_m \\
&= \beta_m^T x^{m_1}(k) + \sum_{k=1}^{N} \alpha_m(k) K_m(x^{m_2}(k), \\
&\quad x^{m_2}(k)) + c_m \quad (20)
\end{aligned}$$

for $m = 1, \ldots, M$. The coefficients can be derived by solving the linear matrix equations such as (19).

Although the MIMO identification problem is solved by treating each channel individually, the linear and nonlinear coupling effects among channels are fully considered in the estimation of channel models, and the model parameters are optimized in terms of a performance in (14), which minimize both the estimation error and weight values. Simulations later show that, although each channel is treated individually, the physically important parameters [i.e., $\beta_m$in (18)–(20)] estimated in each channel model form a spatial-dependent function, which can be estimated accurately by the PM-RLS-SVM.

A. Increment Algorithm

Most LS-SVM algorithms work in an offline manner. This section develops an online LS-SVM algorithm for the identification of the partially linear model. The estimated MIMO partially linear model (20) should be updated with new measurement data at each sampling time. When a new data pair $(x(N+1), Y(N+1))$ is available, the LS-SVM linear matrix equation for the mth channel using $N+1$ and N samples are given respectively as follows:

$$\begin{aligned}
A_{m,N+1}[\beta_m, c_m, \alpha_m, \alpha_m(N+1)]^T &= [0, 0, y_m^N, y_m(N+1)]^T \\
A_{m,N}[\beta_m, c_m, \alpha_m]^T &= [0, 0, y_m^N]^T \quad (21)
\end{aligned}$$

where $y_m^N = (y_m(1), \ldots, y_m(N))$. The relationship between $A_{m,N}$ and $A_{m,N+1}$ is

$$A_{m,N+1} = \begin{bmatrix} A_{m,N} & a_m \\ a_m^T & h_m \end{bmatrix} \quad (22)$$

where $a_m = [x^{m_1(N+1)}; 1; K_m(x^{m_2}(1), x^{m_2}(N+1)); \ldots; K_m(x^{m_2}(N); x^{m_2}(N+1))]$, $h_m = \gamma + K_m(x^{m_2}(N+1), x^{m_2}(N+1))$. The new sampling data can be used to update the model parameters without computing the inverse of the new matrix $A_{m,N+1}$, by adopting the method in [35], as

$$A_{m,N+1}^{-1} = \begin{bmatrix} A_{m,N}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \left[h_m - a_m^T A_N^{-1} a_m \right]^{-1} \begin{bmatrix} A_{m,N}^{-1} a_m \\ -1 \end{bmatrix} \left[a_m^T A_{m,N}^{-1} \; -1 \right]. \quad (23)$$

Note that the computation of the inverse of matrix $A_{m,N+1}$ usually involves an additional computation cost.

B. Sparseness Strategy

This section considers a sparseness strategy for the partially linear system based on pruning error minimization principle [30]. The computation load in the LS-SVM is mainly due to the dimensions of the kernel matrix, which are related to

the number of samples used for regression. The more the number of samples involved in the regression, the worse is the computation complexity. Therefore, pruning strategies should be adopted to reduce the computation complexity while guaranteeing the generalization ability of the model. In general, for an online algorithm, it is better to ignore the sampling data that have the least contribution to the approximation. If such a sample is pruned, the computation load is reduced and the global optimum could still be guaranteed with the rest of samples. With this idea, the pruning error minimization principle is adopted in this paper to prune the training samples which introduce the smallest approximation error after they are omitted.

Consider again the primal optimization problem

$$\begin{aligned} J_m &= \min_{W_m,c_m,e_m(k),\beta_m} \frac{1}{2}W_m^T W_m + \frac{1}{2\gamma}\sum_{t=1}^{N} e_m(k)^2 \\ &= \min_{W_m,c_m,e_m(k),\beta_m} \frac{1}{2}W_m^T W_m + \frac{1}{2\gamma}e_m(1)^2 \\ &\quad + \frac{1}{2\gamma}e_m(2)^2 + \cdots + \frac{1}{2\gamma}e_m(N)^2 \end{aligned} \tag{24}$$

with equality constraints

$$y_m(k) = \beta_m^T x^{m_1}(k) + W_m^T \varphi_m(x^{m_2}(k)) + c_m + e_m(k) \tag{25}$$

for $k = 1, \ldots, N$. In the regression, the importance of $e_m(k)$ is determined by γ. For $e_m(i)$, if we let γ become smaller ($(1/\gamma)$ becomes larger), it will increase the effect of the sampling data $(x(i), Y(i))$. Furthermore, if we let $\gamma \to \infty$ ($(1/\gamma) \to 0$), $e_m(i)$ will no longer have any effect in the ridge regression.

Using the estimated model (20), the output of channel m with $N+1$ kernels (i.e., considering $N+1$ sampling data) with respect to the training sample j can be calculated as

$$\begin{aligned} & y_m^{N+1}(x(j)) \\ &= \beta_{m,N+1}^T x^{m_1}(j) + \sum_{k=1,k\neq j}^{N+1} \alpha_m^{N+1}(k) K_m(x^{m_2}(k), x^{m_2}(j)) \\ &\quad + \alpha_m^{N+1}(j) K_m(x^{m_2}(j), x^{m_2}(j)) + c_m^{N+1}. \end{aligned} \tag{26}$$

The output of channel m with N kernels with respect to the training sample j can be calculated as

$$\begin{aligned} y_m^N(x(j)) &= \beta_{m,N}^T x^{m_1}(j) \\ &\quad + \sum_{k=1,k\neq j}^{N+1} \alpha_m^N(k) K_m(x^{m_2}(k), x^{m_2}(j)) + c_m^N. \end{aligned} \tag{27}$$

Then by subtracting the two equations above, the introduced error of the mth output channel for sample j, when j is omitted, is given by

$$\begin{aligned} D_m(x(j)) &= y_m^{N+1}(x(j)) - y_m^N(x(j)) \\ &= (\beta_{m,N+1}^T - \beta_{m,N}^T) x^{m_1}(j) \\ &\quad + \sum_{k=1,k\neq j}^{N+1} (\alpha_m^{N+1}(k) - \alpha_m^N(k)) K_m(x^{m_2}(k), x^{m_2}(j)) \\ &\quad + \alpha_m^{N+1}(j) K_m(x^{m_2}(j), x^{m_2}(j)) \\ &\quad + c_m^{N+1} - c_m^N. \end{aligned} \tag{28}$$

Since the computation complexity is directly dependent on the number of samples, in order to reduce it while guaranteeing the generalization, in the case of $N+1$ samples, one needs to prune one sample that carries the least information compared to the other samples. Intuitively, the sample that introduces the smallest approximation error after being omitted could be pruned. Note that the introduced error for the mth output channel is determined by the difference of coefficients β_m, $\alpha_m(k)$'s and c_m. Now, compute the approximation error introduced by sample j. Let the regularization constant γ with respect to $e_m(j)$ tend to infinity, which implies that a parameter $\lambda(\lambda$ tends to infinity) is added to γ with respect to $e_m(j)$. This leads to the following linear matrix equation:

$$\begin{bmatrix} 0 & 0 & X_{m_1}^T \\ 0 & 0 & 1^T \\ X_{m_1} & 1 & \Omega_m^{N+1} + \gamma I + V_j \end{bmatrix} \cdot \begin{bmatrix} \beta_m \\ c_m \\ \alpha_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ y_m \end{bmatrix} \tag{29}$$

where $V_j = diag\{0, \ldots, \lambda, \ldots, 0\}$, $X_{m_1} = [x^{m_1}(1)^T, x^{m_1}(2)^T, \ldots, x^{m_1}(N+1)^T]$, $y_m = [y_m(1), y_m(2), \ldots, y_m(N+1)]$. If $\lambda = 0$, the equation is just the case of $N+1$ samples, if $\lambda \to \infty$, the sample j is ignored in the regression. We can obtain (see Appendix) the introduced error after sample j is omitted as

$$D_m(x(j)) = \frac{\alpha_m^{N+1}(j)}{[A_{m,N+1}^{-1}]_{jj}} \tag{30}$$

where $[A_{m,N+1}^{-1}]_{jj}$ represents the $N_{m_1}+1+j$ diagonal element of the inverse of $A_{m,N+1}^{-1}$. For the MIMO partially linear system (11), there are M output channels. Therefore, the introduced error for all the channels should be considered to find the sample to prune. To this aim, the following criterion is proposed:

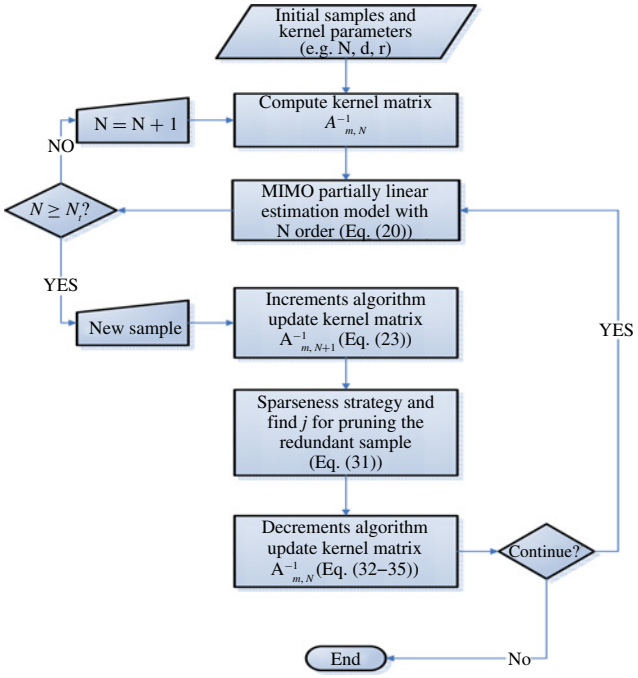
$$\arg_j \min \sum_{m=1}^{M} |D_m(x(j))| = \arg_j \min \sum_{m=1}^{M} \left| \frac{\alpha_m^{N+1}(j)}{[A_{m,N+1}^{-1}]_{jj}} \right|. \tag{31}$$

With this pruning method, the kernel with the least contribution would be removed to keep the parsimony and generalization ability of the estimation model.

The commonly used pruning strategies for LS-SVM are the moving window strategy [36] and the least Lagrange multiplier strategy [37]. The former usually deletes the oldest sample by supposing the newer one contains more information about the system. The latter tries to remove the sample with the smallest Lagrange multiplier. However, both have no solid theoretical proof and may not guarantee the generalization ability after the sample is pruned. However, based on the pruning error minimization principle, the sparseness strategy given in this section can theoretically guarantee the minimum estimation error and probably the best generalization of the model.

C. Decrement Algorithm

The increment algorithm above demonstrates how to update the estimate model when a new sample is available without computing the inverse of the updated kernel matrix. To construct an online algorithm, after the new sample is available, the redundant sample in the new training dataset that is



selected by the sparseness strategy needs to be pruned. Similar to the increment algorithm, it is better to avoid computing the inverse of the matrix. Thus, $A^{-1}_{m,N}$ that will appear after the redundant sample is pruned should be updated from $A^{-1}_{m,N+1}$.

According to the proposed sparseness strategy, any sample in the training set could be pruned. If sample j is pruned, for the mth output channel, it means that $A_{m,N}$ is constructed by deleting the $N_{m_1}+1+j$th row and $N_{m_1}+1+j$th column of $A_{m,N+1}$. For $A^{-1}_{m,N+1}$ and $A^{-1}_{m,N}$, denote the inverse of $A_{m,N+1}$ by

$$A^{-1}_{m,N+1} = \begin{bmatrix} A^{-1}_{m,N+1}(1,1) & a_m(1,j) & A^{-1}_{m,N+1}(1,2) \\ a^T_m(1,j) & a^*_m & a^T_m(2,j) \\ A^{-1}_{m,N+1}(2,1) & a_m(2,j) & A^{-1}_{m,N+1}(2,2) \end{bmatrix} \quad (32)$$

and

$$\overline{A}^{-1}_{m,N+1} = \begin{pmatrix} A^{-1}_{m,N+1}(1,1) & A^{-1}_{m,N+1}(1,2) \\ A^{-1}_{m,N+1}(2,1) & A^{-1}_{m,N+1}(2,2) \end{pmatrix} \quad (33)$$

$$H_{m,N+1} = \begin{pmatrix} a_m(1,j) \\ a_m(2,j) \end{pmatrix} \quad (34)$$

where $[a_m(1,j), a^*_m, a_m(2,j)]^T$ and $[a^T_m(1,j), a^*_m, a^T_m(2,j)]$ represent the $N_{m_1}+1+j$ column and $N_{m_1}+1+j$ row of $A^{-1}_{m,N+1}$. The inverse of $A_{m,N}$ can be obtained by

$$A^{-1}_{m,N} = \overline{A}^{-1}_{m,N+1} - \frac{1}{a^*_m} H_{m,N+1} H^T_{m,N+1}. \quad (35)$$

Then, $A^{-1}_{m,N}$ can be updated from $A^{-1}_{m,N+1}$ without the inverse of $A_{m,N}$.

Generally speaking, a specific spatiotemporal system corresponds to a specific lattice structure and difference relationship. After model transformations, it would become a MIMO partially linear model. The proposed algorithm can be used to reproduce online the dynamic behaviors of spatiotemporal dynamical systems and also to characterize the physical characteristics of the system as mentioned before. The algorithm is summarized in the figure top. N_t is the threshold number of the training data that are used for regression.

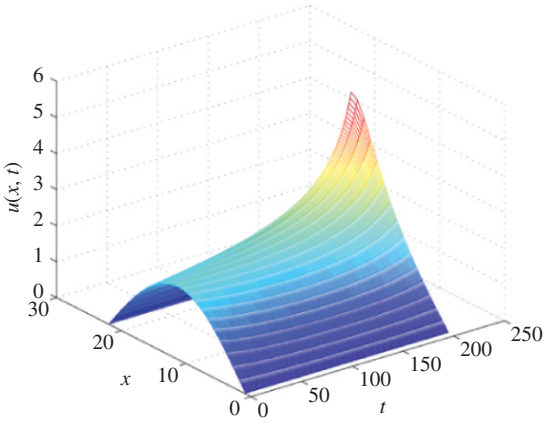


Fig. 1. Actual data $u(x,t)$.

V. Simulations and Discussions

A. Example A: Kolmogorov–Petrovskii–Piskunov Equation

Let $u(x,t)$ represent the temperature distribution of an insulated rod and assume that there is a nonlinear heat source of strength $0.1u(x,t)(5u(x,t)-1)(5u(x,t)+1)$ (Kolmogorov–Petrovskii–Piskunov equation [2]). The nonlinear equation to be studied has the form

$$\frac{\partial u}{\partial t} = (1+\sin^3(2\pi x))\frac{\partial^2 u}{\partial x^2} + 0.1u(x,t)(25u^2(x,t)-1) \quad (36)$$

with the initial and boundary conditions

$$u(x,0) = 2\sin(\pi x), \quad u(0,t) = 0, \quad u(1,t) = 0. \quad (37)$$

This system can be transformed into a lattice dynamical system following Section II and then a MIMO partially linear system when both the space and time are properly discretized.

In identification, the space domain is sampled at 22 equally spaced points over [0, 1]. Therefore, a 20-D MIMO partially linear model is established. The time domain is sampled at 200 equally spaced points over [0, 0.1]. Thus 200 data points for each dimension is generated. The actual data with space step $\triangle x = 1/21$ and time step $\triangle t = 0.1/200$ is plotted in Fig. 1.

The commonly used positive kernel functions are the linear kernel ($K(x_i,x_j) = x^T_i x_j$), the polynomial kernel ($K(x_i,x_j) = (x^T_i x_j + r)^d$, where d is polynomial degree and r is tuning parameter), and the Gaussian kernel ($K(x_i,x_j) = \exp(-||x_i - x_j||^2_2/\sigma^2)$, where σ is the bandwidth). For each dimension, by solving the corresponding matrix equation, the coefficients of the linear part for the regression model can be derived. Then, the space-dependent coefficient $a(x)$, which is a nonlinear function representing the physical property of the heat transition medium, can be obtained by proper scaling. Without loss of generality, the corresponding data of $a(x)$ is taken at $k = 21$, after a threshold number of

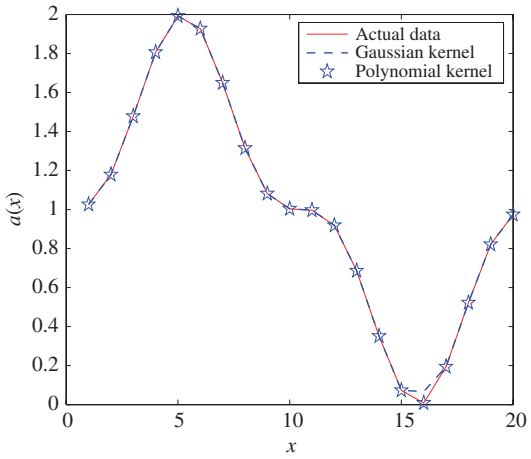


Fig. 2. Estimations for $a(x) = 1 + \sin^3(2\pi x)$.

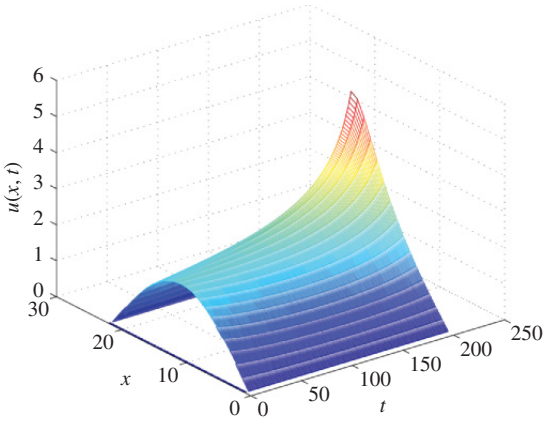


Fig. 5. Model prediction output using a Gaussian kernel.

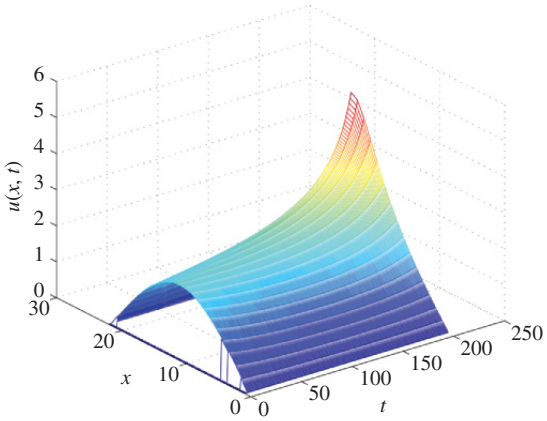


Fig. 3. Model prediction output using a polynomial kernel.

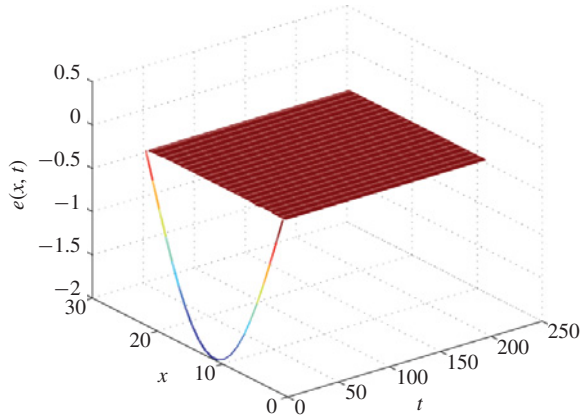


Fig. 6. Model prediction error using the Gaussian kernel.

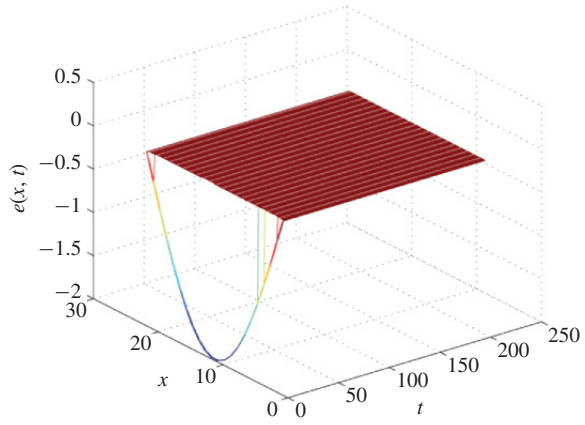


Fig. 4. Model prediction error using a polynomial kernel.

samples are available for the regression model to avoid the transient response (the threshold number is taken as 20 given heuristically in this example). The result is shown in Fig. 2. The model prediction and prediction error using polynomial kernel functions are shown in Figs. 3 and 4, and those using Gaussian kernel functions are given in Figs. 5 and 6. In the first case, the proposed online algorithm leads to 20 samples that are finally selected out as (123, 142, 147, 149, 150, 156, 168, 182, 183, 184, 187, 189, 191, 192, 193, 194, 195, 196, 199, 200) and the polynomial kernel function is chosen with parameters $d = 3$, $r = 1$, and $\gamma = 0.001$. In the second case, by using the online algorithm, 20 samples are finally selected out (169, 172, 173, 174, 175, 178, 179, 180, 181, 182, 183, 185, 190, 191, 192, 193, 194, 196, 199, 200) and the Gaussian kernel function is chosen with parameters $\sigma = 10$ and $\gamma = 0.001$. It can be seen that the proposed algorithm tracks the dynamics very well and the identified model can accurately reproduce the behavior of the spatiotemporal system after a short period of transient response (about five sampling times in Figs. 4 and 6). With proper scaling, the nonlinear $a(x) = 1 + \sin^3(2\pi x)$ is precisely fitted.

In the simulation, at a fixed dimension i, each corresponding coefficient of the linear part is available to obtain the coefficient $a(i)$. Therefore, the average value of the coefficients are used (actually, the coefficients of the linear part are slightly different). Satisfactory results could be obtained with fewer samples (20 training data are used here).

The selection of kernel parameters may have some influence on the identification performance [25], [34], [38]. For the linear and Gaussian kernels, it is efficient to identify a more complex nonlinear process by using a larger degree d or a smaller bandwidth σ^2. But it may result in overfitting problems, when the nonlinearity of the process is relatively weak.

TABLE I
IDENTIFICATION RESULTS WITH DIFFERENT PARAMETERS UNDER DIFFERENT CONDITIONS

Kernel	γ	(d, r)	σ	MSE	RMSE	Samples
Poly	10^{-3}	(3, 1)		5.3×10^{-6}	4.1×10^{-6}	20
Poly	10^{-3}	(4, 1)		4.8×10^{-6}	4.4×10^{-6}	20
Poly	10^{-3}	(5, 1)		5.5×10^{-5}	2.8×10^{-5}	10
RBF	10^{-6}		10	2.8×10^{-6}	7.4×10^{-6}	20
RBF	10^{-6}		1	4.5×10^{-8}	8.1×10^{-8}	20
RBF	10^{-3}		10	4.0×10^{-5}	7.3×10^{-5}	20
RBF	10^{-3}		1	2.7×10^{-8}	5.9×10^{-8}	20

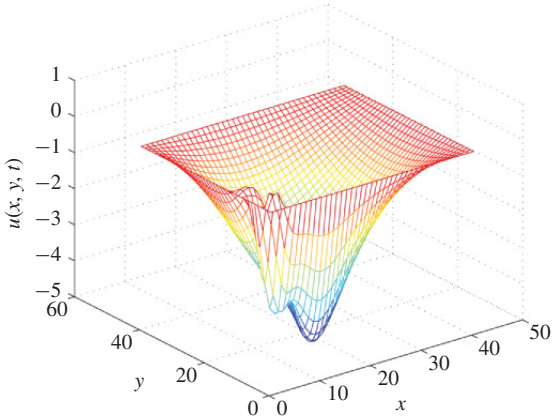


Fig. 7. Model prediction output $u(x, y, t)$ using a polynomial kernel at $t = 2$.

When a smaller degree d and a larger bandwidth σ^2 are used, overfitting problems may be avoided, but it is difficult to trace a complex nonlinear process. The regularization parameter γ can be preselected, a larger γ could help to derive a more complex model, and vice versa. Therefore, some prior information about the system nonlinearity may be helpful for the selection of kernel parameters.

To demonstrate the proposed algorithm, two performance indices are introduced, i.e., the mean square error (MSE$=\sqrt{\sum_{i=1}^{M}\sum_{k=7}^{N}(\bar{u}(i,k)-u(i,k))^2/MN}$), and the relative mean square error (RMSE$=\sqrt{\sum_{i=1}^{M}\sum_{k=7}^{N}(\bar{u}(i,k)-u(i,k)/u(i,k))^2/MN}$), where $\bar{u}(i,k)$ is used to denote the predicted output of $u(i,k)$. Note that the MSE and RMSE are calculated without the data with $N < 7$ to remove the transient response effects. The Gaussian kernel function and polynomial kernel function are utilized with different kernel parameter (γ, σ) and (γ, d, r) (see Table I). Although the selection of kernel parameters does affect the results, the identification performance is slightly different as shown when the values of the parameters are in a reasonable range as discussed before.

B. Example B: Nonlinear Klein–Gordon Equation

Another example is used to show the application of the proposed method to a 2-D problem. Consider the following

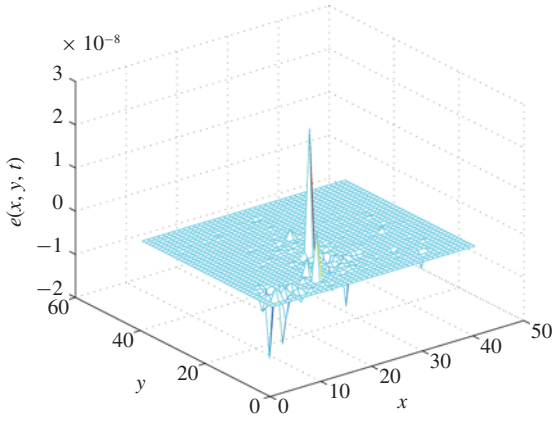


Fig. 8. Model prediction error using polynomial kernel at $t = 2$.

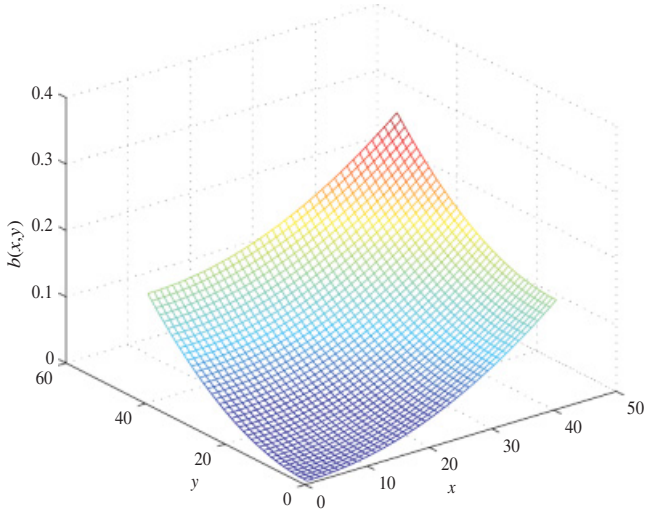


Fig. 9. Estimations for $b(x, y)$ using a polynomial kernel.

nonlinear Klein–Gordon equation:

$$\frac{\partial^2 u}{\partial t^2} = \left(x^2 + y^2\right)\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + 0.2u^2(x, y, t) \quad 0 \le x \le 2, \quad 0 \le y \le 2 \tag{38}$$

with boundary conditions and initial conditions

$$\begin{aligned} u(x, 0, t) &= 0, \quad u(x, 2, t) = 0, \\ u(0, y, t) &= 0, \quad u(2, y, t) = 0, \\ u(x, y, 0) &= 0.1 \sin(\pi x) \sin\left(\frac{\pi y}{2}\right), \\ \frac{\partial u}{\partial t}(x, y, 0) &= 0. \end{aligned} \tag{39}$$

Following Section II, this system can be transformed into a MIMO partially linear model with proper discretization. To apply the proposed algorithm, the space domains along the x- and y-axes are sampled evenly with 21 points over $[0, 2]$. A 361-D MIMO partially linear model is then established. The time domain is sampled evenly with 200 points over $[0, 2]$. Two hundred data points for each dimension are generated.

By using polynomial kernel functions with parameters $d = 3$, $r = 1$, and $\gamma = 10^{-9}$, 50 samples are selected out. The model prediction and prediction error for $u(x, y, t)$ at

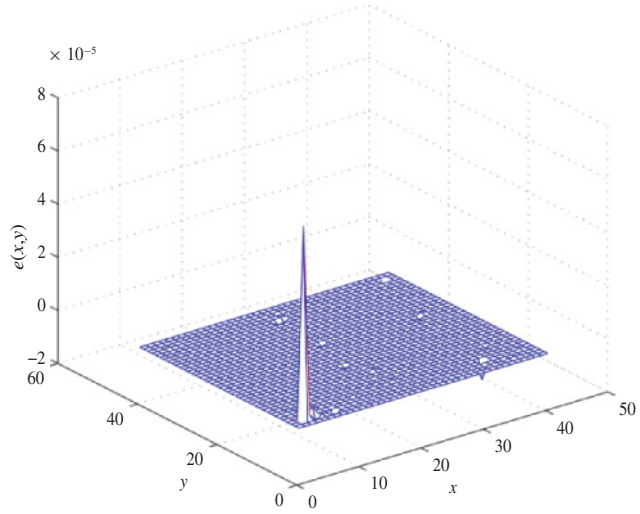


Fig. 10. Estimation errors for $b(x, y)$ using a polynomial kernel.

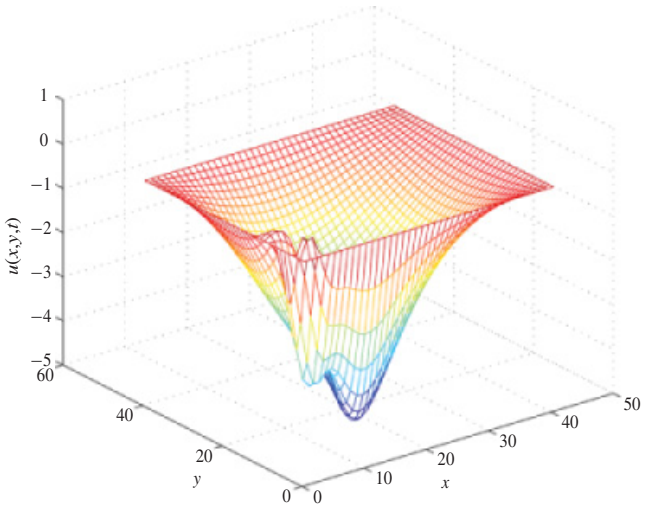


Fig. 11. Model prediction output using the Gaussian kernel at $t = 2$.

TABLE II
IDENTIFICATION RESULTS WITH DIFFERENT PARAMETERS UNDER DIFFERENT CONDITIONS

Kernel	γ	(d, r)	σ	MSE	RMSE	Samples
Poly	10^{-9}	(2, 1)		1.1×10^{-6}	1.2×10^{-3}	50
Poly	10^{-9}	(3, 1)		1.8×10^{-6}	2.3×10^{-3}	50
Poly	10^{-9}	(5, 1)		3.8×10^{-6}	9.6×10^{-3}	30
RBF	10^{-9}		10	2.03×10^{-4}	3.5×10^{-3}	50
RBF	10^{-9}		100	7.6×10^{-6}	1.2×10^{-3}	50
RBF	10^{-9}		100	2.3×10^{-4}	3.3×10^{-3}	30

$t = 0.01 \times 200 = 2$ are plotted in Figs. 7 and 8, respectively. The estimated values and their errors for $b(x, y)$ are shown in Figs. 9 and 10. The model prediction and prediction error for $u(x, y, t)$ at $t = 0.01 \times 200 = 2$ using Gaussian kernel functions are given in Figs. 11 and 12. The estimated values and their errors for $b(x, y)$ are shown in Figs. 13 and 14. The Gaussian kernel function is chosen with parameters $\sigma = 10$ and $\gamma = 10^{-9}$. The results with different kernel functions and parameters are summarized in Table II.

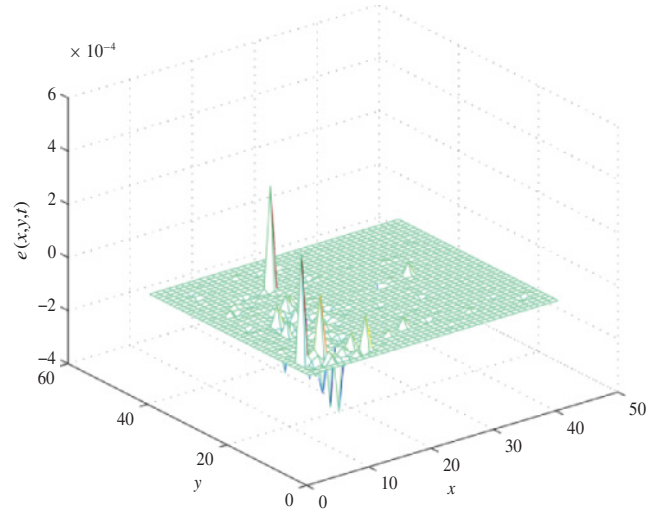


Fig. 12. Model prediction error using the Gaussian kernel at $t = 2$.

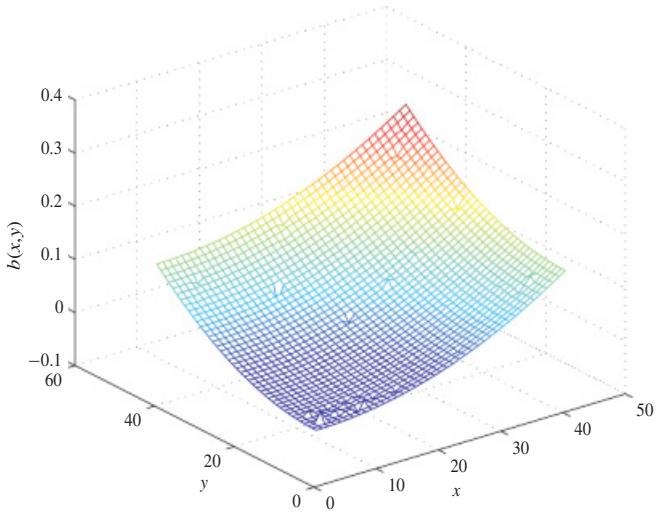


Fig. 13. Estimations for $b(x, y)$ using the Gaussian kernel.

These results still demonstrate the good performance of the proposed method. Note that the estimation error on the node near the original point is relatively large (a peak in Fig. 10). The reason could be that $b(x, y)$ is actually very small on this node, which may consequently result in bigger rounding error in the regression. Moreover, it should be noted that multiple step ahead prediction using the proposed method can also be achieved with reasonably small prediction error, since one step ahead prediction is very accurate (Figs. 4, 6, and 8).

C. Comparison Results

Compared to the existing methods developed for the identification of spatiotemporal systems in [14]–[17], the proposed method utilizes the known structural information about the system so that the nonlinear physical characteristics of the original system can be estimated simultaneously. However, the FOLS methods in [14]–[17] produce only an NARX model or black-box model having little physical link. The FOLS algorithms are only offline ones and involve repetitively using

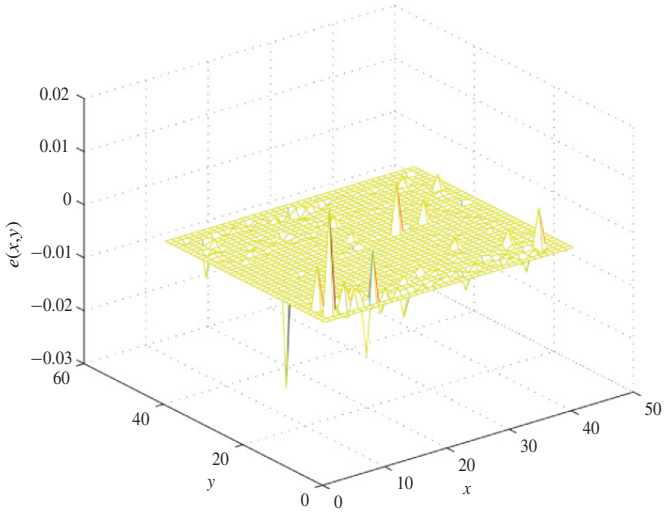


Fig. 14. Estimation errors for $b(x, y)$ using the Gaussian kernel.

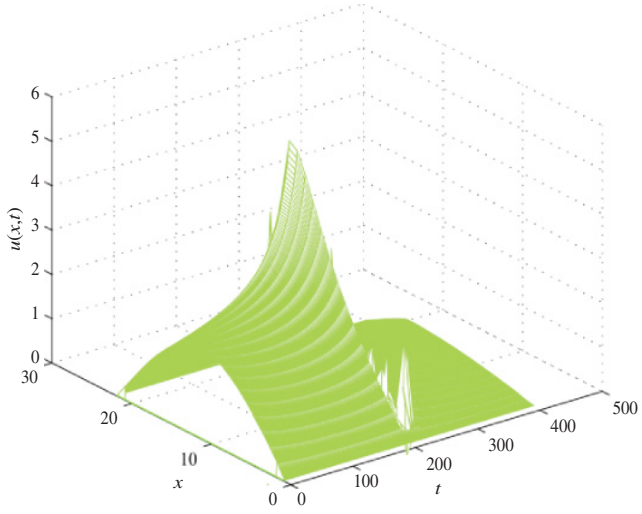


Fig. 16. Model prediction output.

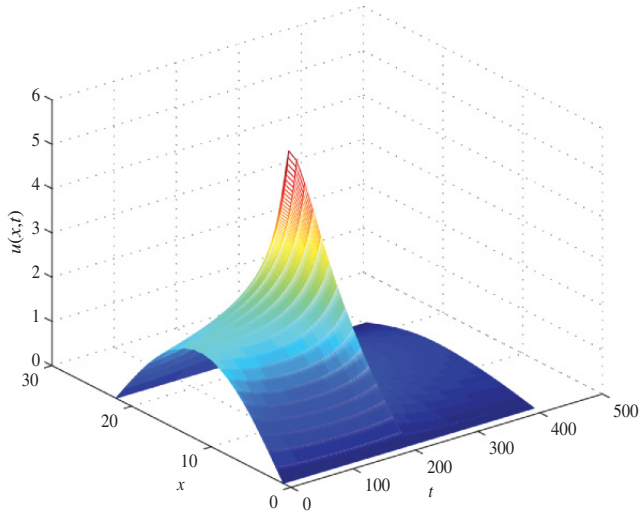


Fig. 15. Actual data $u(x, t)$ (an abrupt change at time 0.1).

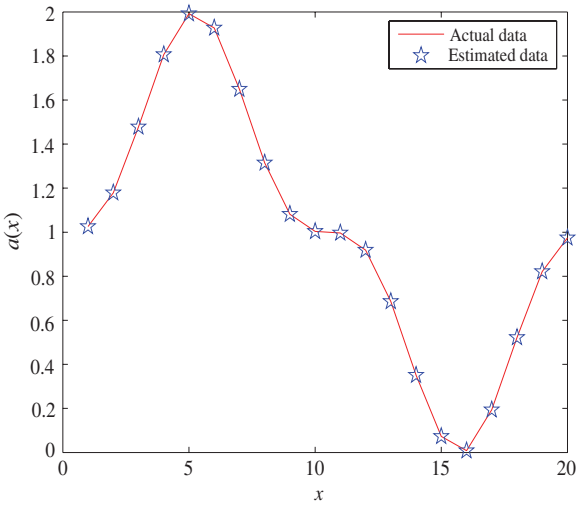


Fig. 17. Estimations for $a(x)$ at time 0.05.

the Gram–Schmidt orthogonalization process for each new sample whose complexity is known as $O(N_s^3)$ [14], [19], [20], [39], where N_s is the number of samples used for regression. The proposed method is a recursive version of the LS-SVM with recursive update of the matrix inversions, and the computation complexity of the offline LS-SVM is estimated as $O(N_s^2)$ [34], [33]. Therefore, the complexity of the PM-RLS-SVM must be better than the FOLS in [14]–[17] and can be used for online estimation.

Note that several kernel learning methods such as kernel least mean squares (KLMS) and kernel recursive least squares (KRLS) were proposed recently [40]–[43]. Although these algorithms have efficient sparseness and update strategies, none employs pruning strategy.

The pruning strategy in the PM-RLS-SVM can remove the sample with the least contribution to the model, and therefore can maintain the parsimony of the model and trace the changing dynamics of the system (if any) quickly. Without the pruning strategy, the performance would be different. For better understanding, consider a special cases shown in Fig. 15, where the system dynamics undergo significant changes at time t^*. In order to accurately predict the system behavior after t^*, obviously the old samples measured before t^* would take a very limited or even negative contribution and thus should be pruned from the regression. The PM-RLS-SVM can effectively achieve this and quickly keep up with the changing of the system dynamics. However, the KLMS and KRLS algorithms cannot remove the old useless information and thus would bring larger errors or biased estimation for system model and system states.

Moreover, the proposed method in this paper can accurately estimate the model parameters that characterize system physical properties and are nonlinear functions of space variables. The KLMS and KRLS algorithms are developed in the feature space aiming at minimizing the estimation error only [compare with (14)], and the KRLS additionally requires the approximate linear dependence condition [40]–[43]. Therefore, they may not be directly extended to the identification of partially linear models with capability of estimating important physical parameters simultaneously.

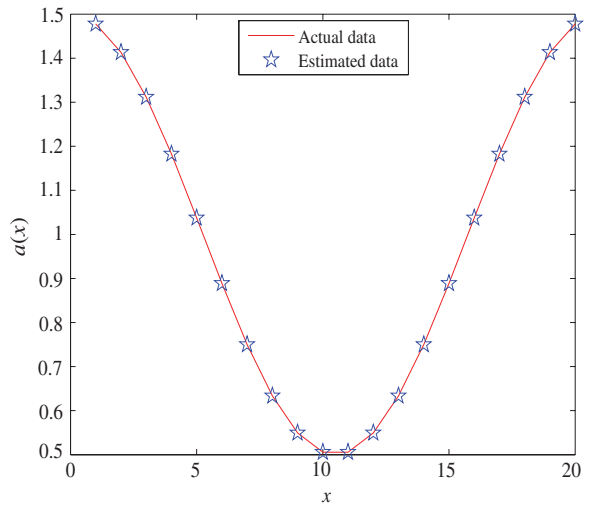


Fig. 18. Estimations for $a(x)$ at time 0.15.

TABLE III
IDENTIFICATION RESULTS USING DIFFERENT ONLINE ALGORITHMS

Method	(d, r)	γ	η	ν	MSE	RMSE
PM-RLS-SVM	(3, 1)	0.001			1.78×10^{-5}	1.47×10^{-5}
KLMS	(3, 1)		0.5		7.9×10^{-2}	8.1×10^{-2}
KRLS	(3, 1)			0.05	1.27×10^{-2}	3.2×10^{-2}
KLLS	(3, 1)				larger	larger

To illustrate the advantages of the PM-RLS-SVM algorithm in tracking changing nonlinear dynamics and changing physical characteristics, consider Example A again with similar discretization in the time interval [0, 0.2] instead of [0, 0.1]. At $t = 0.1$ (i.e., the sampling time 201), the nonlinear term $f(u(x,t))$ is designed to change from $f(u) = 0.1u(25u^2 - 1)$ to $f(u) = 5u - 15u^2$, the parameter $a(x)$ is designed to change from $a(x) = 1 + \sin^3(2\pi x)$ to $a(x) = 1 + 0.5\cos(2\pi x)$, and the input $u(x,t)$ is also changed at the same time. That is, the boundary condition $u(x, 0.1) = 4x(1 - x)$, $0 \le x \le 1$ (see Fig. 15). In simulations, the polynomial kernel function is chosen with $d = 3$, $r = 1$. If the prediction error exceeds 0.5, then take it as 0.5 for convenience in visual illustration. The prediction output and errors with the proposed algorithm are given in Figs. 16 and 19. The data of $a(x)$ taken at time 0.05 and 0.15 is shown in Figs. 17 and 18. It can be seen that after a short period (about 15 sampling times) of transient effects, the dynamics of the system can be predicted very well, and the change of $a(x)$ is estimated accurately. The prediction errors with the least Lagrange multiplier strategy (KLLS) [37] is shown in Fig. 20, where γ is also chosen to 0.001, and with KLMS and KRLS given in Figs. 21 and 22, where the learning step η for KLMS is chosen as 0.5 and the ALD condition parameter ν for KRLS is chosen as 0.05 for as good a performance as possible. Large estimation errors are observed even after transient response. The MSE and RMSE using these algorithms are shown in Table III, where the transient effects are removed in computing the MSE and RMSE.

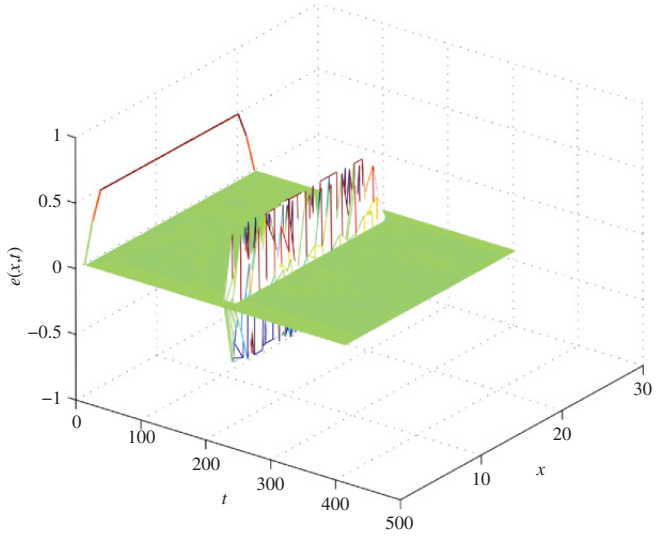


Fig. 19. Model prediction error (an abrupt change at time 0.1).

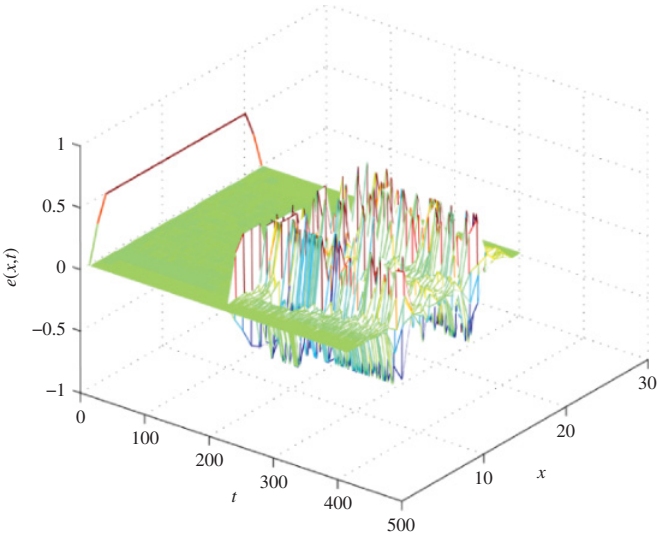


Fig. 20. Model prediction error using the least Lagrange multiplier strategy.

Compared to the KLMS and KRLS in [40]–[43] and the least Lagrange multiplier strategy [37], the PM-RLS-SVM demonstrates obviously a better performance in tracing the changing dynamics of the system in this example.

Furthermore, it should be noted that this could be the first attempt to use a recursive LS-SVM method to tackle the (online) estimation problems of distributed parameter systems from the perspective of a partially linear model, although the LS-SVM methods have already been applied to identify partially linear models recently such as the PL-SVM in [33]. Compared to the PL-SVM, the PM-RLS-SVM provides an efficient recursive version of the LS-SVM algorithm (i.e., an increment algorithm and a decrement algorithm with a well-proven sparseness strategy and efficient update method for matrix inversion), copes with a MIMO partially linear model, which is reflected in the coupling effects in the linear and nonlinear parts of each channel model (7)–(20), and can

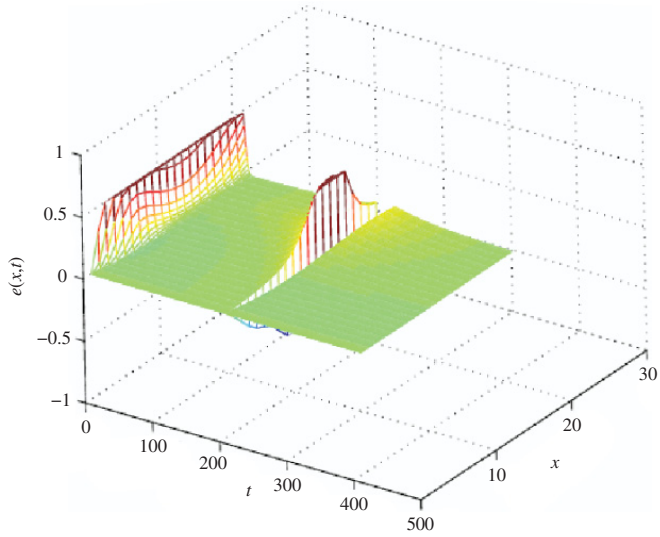


Fig. 21. Model prediction error using KLMS.

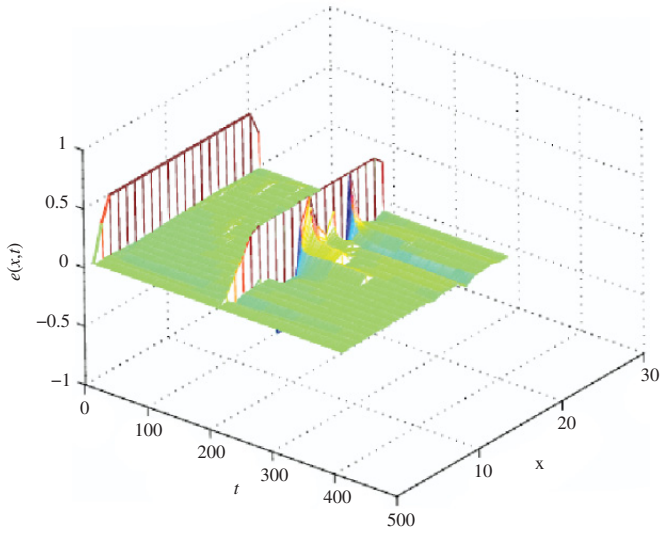


Fig. 22. Model prediction error using KRLS.

estimate important physical parameters accurately (a spatio-dependent function involved in all channel models).

D. Further Discussions

Generally speaking, many physical phenomena can be described by spatiotemporal models using PDEs. With some classic discretization methods, these models can always be transformed into lattice dynamical systems and then into partially linear models which can be identified with the PM-RLS-SVM. The kernel-based learning methods as demonstrated in this paper could be a promising and very powerful tool to investigate the analysis and estimation problems of distributed parameter systems and thus provide a useful insight into data-based optimization and control of PDEs [44], [45].

In practice, the measurement noise is inevitable. Considering (36), suppose the measurement data of $u(x, y, t)$ are perturbed by Gaussian noise $e(t)$. That is, u^* is obtained and

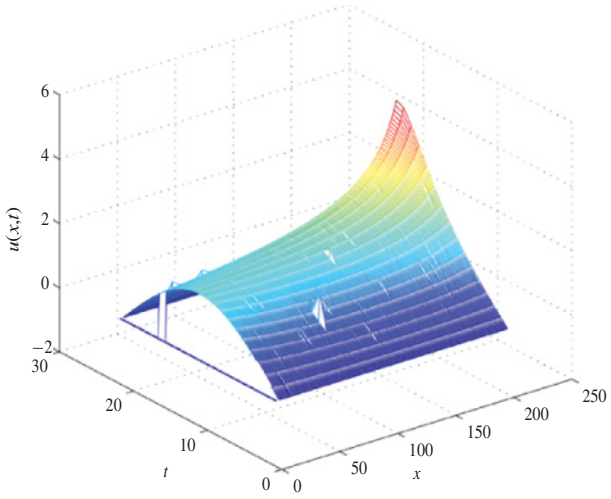


Fig. 23. Model prediction output.

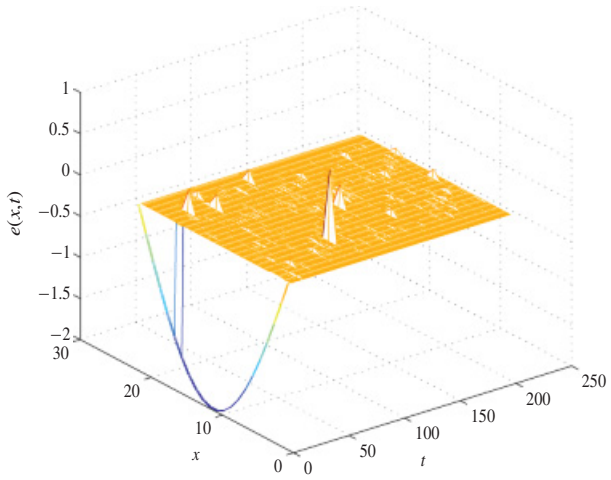


Fig. 24. Model prediction error (subject to measurement noise).

used in regression with

$$u^*(x, y, t) = u(x, y, t) + e(t). \tag{40}$$

The nonlinear term of the model for Example A contains the state $u(x, y, t)$ $(f(u))$. Substituting $u(x, y, t)$ by $u^*(x, y, t)$, the regression model can be generally written as

$$u^*(t + 1) = f^*(u^*(t) + e(t)) + e(t + 1). \tag{41}$$

The measurement noise actually enters into the regression nonlinearly as a process noise, which is a difficult problem in nonlinear system identification. A little noise may have great influence on the identification algorithm and result in large estimation errors.

Considering (36), a simulation result is given to demonstrate the proposed algorithm in treating noisy data. The data are corrupted by independent Gaussian noise with variance of 0.001. The polynomial kernel function is chosen with $d = 3$, $r = 1$, and $\gamma = 0.001$. Twenty samples are finally selected. The results are given in Figs. 23 and 24, showing that the noise does have an influence on identification, but the estimated dynamics of the system still matched well. For spatiotemporal dynamical systems with suitable nonlinearity, satisfactory

performance with the proposed identification algorithm could still be obtained in noisy environments.

Alternatively, spatiotemporal systems perturbed by noise may be studied based on stochastic PDEs [46]–[49]. A simple example widely applied in neurophysiology [48] is given by

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) = \frac{\partial^2 u}{\partial x^2}(x,t) - bu(x,t) + f(x,t) \\ \qquad\qquad + \frac{\partial^2 W}{\partial t \partial x}(x,t), t > 0, \\ u(x,0) = \xi(x), \quad 0 \le x \le 1, \\ u(0,t) = u(1,t) = 0, \quad t \ge 0 \end{cases} \tag{42}$$

where $(\partial^2 W/\partial t \partial x)(x,t)$ denotes the space–time white noise, b is a constant. There are many different stochastic PDEs for different physical phenomena and different noise (e.g., levy noise). Therefore, the application and extension of the identification method proposed in this paper to stochastic spatiotemporal systems are interesting topics for further investigation.

VI. Conclusion

A systematic identification method for nonlinear spatiotemporal systems was developed by using the LS-SVM method from a perspective of partially linear models. The spatiotemporal dynamical systems were formulated into a general MIMO partially linear system, and an online algorithm was therefore proposed based on the LS-SVM and a well-proven pruning error minimization principle. Because the proposed method takes advantage of the prior structural information of the samples to determine the structure of regression models and corresponding regressors, this results in reduced computation complexity and powerful characterization of the dynamic and nonlinear characteristics of the physical or structural properties of the underlying system. The latter is of particular significance in the analysis and design of nonlinear spatiotemporal systems, which now is under further study. The method can also be extended to a more general case with stochastic process noise and the control and optimal design of distributed parameter systems. These will be dealt with in future studies.

Appendix

Calculation of the Introduced Error

Denote $(0,0,y_m) = \delta_m$, $p = [0,0,0,\sqrt{\lambda},0]^T$, $I_{m,j}$ the column vector of size $N+2+N_{m_1}$ filled with 0 except element $N_{m_1}+1+j$ which is equal to 1, and

$$A_{m,N+1,\lambda} = \begin{pmatrix} 0 & 0 & X_{m_1}^T \\ 0 & 0 & 1 \\ X_{m_1} & 1 & \Omega_m^{N+1} + \gamma I + V_j \end{pmatrix}. \tag{A1}$$

The kernel matrix can be represented by $A_{m,N+1} + pp^T$. With the method in [35]

$$\begin{aligned} A_{m,N+1,\lambda}^{-1} &= [A_{m,N+1} + pp^T]^{-1} \\ &= A_{m,N+1}^{-1} - \frac{A_{m,N+1}^{-1} pp^T A_{m,N+1}^{-1}}{1 + p^T A_{m,N+1}^{-1} p}. \end{aligned} \tag{A2}$$

Set $z_m = A_{m,N+1}^{-1}\delta_m$ and $z_{m,\lambda} = A_{m,N+1,\lambda}^{-1}\delta_m$, then

$$\begin{aligned} \triangle z_m &= z_m - z_{m,\lambda} = A_{m,N+1}^{-1}\delta_m - A_{m,N+1,\lambda}^{-1}\delta_m \\ &= \left(\frac{A_{m,N+1}^{-1} pp^T A_{m,N+1}^{-1}}{1 + p^T A_{m,N+1}^{-1} p} \right) \delta_m \\ &= \left(\frac{\lambda A_{m,N+1}^{-1} I_{m,j} I_{m,j}^T A_{m,N+1}^{-1}}{1 + \lambda I_{m,j}^T A_{m,N+1}^{-1} I_{m,j}} \right) \delta_m. \end{aligned} \tag{A3}$$

Taking $\lambda \to \infty$ gives

$$\lim_{\lambda \to \infty} \triangle z_m = \left(\frac{A_{m,N+1}^{-1} I_{m,j} I_{m,j}^T A_{m,N+1}^{-1}}{I_{m,j}^T A_{m,N+1}^{-1} I_{m,j}} \right) \delta_m. \tag{A4}$$

Note that

$$\begin{aligned} D_m(x(j)) &= (\beta_{m,N+1}^T - \beta_{m,N}^T) x^{m_1}(j) \\ &\quad + \sum_{t=1, k \ne j}^{N+1} (\alpha_m^{N+1}(t) - \alpha_m^N(t)) K_m(x^{m_2}(t), x^{m_2}(j)) \\ &\quad + \alpha_m^{N+1}(j) K_m(x^{m_2}(j), x^{m_2}(j)) + c_m^{N+1} \\ &\quad - c_m^N \end{aligned} \tag{A5}$$

which is equal to the product of the difference of the solutions and the $N_{m_1}+1+j$ row of kernel matrix $A_{m,N+1}$, then we have $D_m(x(j)) = (\alpha_m^{N+1}(j)/[A_{m,N+1}^{-1}]_{jj})$.

Acknowledgment

The authors would like to thank the reviewers for their constructive comments and suggestions.

References

[1] L. Debnath, *Nonlinear Water Waves*. Boston, MA: Academic, 1994.
[2] L. Debnath, *Nonlinear Partial Differential Equations for Scientists and Engineers*. Boston, MA: Birkhäuser, 2005.
[3] D. L. Russell, "Controllability and stabilizability theory for linear partial differential equations: Recent progress and open questions," *SIAM Rev.*, vol. 20, no. 4, pp. 639–739, Oct. 1978.
[4] R. E. Showalter, *Monotone Operators in Banach Space and Nonlinear Partial Differential Equations*. Providence, RI: AMS, 1996.
[5] L. Hörmander, "On the existence of real analytic solutions of partial differential equations with constant coefficients," *Invent. Math.*, vol. 21, no. 3, pp. 151–182, 1973.
[6] K. Deimling, *Nonlinear Functional Analysis*. Berlin, Germany: Springer-Verlag, 1985.
[7] R. A. Adams and J. F. Fournier, *Sobolev Spaces*. Boston, MA: Academic, 1975.
[8] W. F. Ames, *Numerical Methods for Partial Differential Equations*. Boston, MA: Academic, 1977.
[9] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge, U.K.: Cambridge Univ. Press, 1987.
[10] A. Quartevoni and A. Valli, *Numerical Approximation of Partial Differential Equations*. New York: Springer-Verlag, 1997.
[11] S. N. Chow, *Lattice Dynamical Systems* (Lecture Notes in Mathematics). New York: Springer-Verlag, 2003.
[12] T. D. Martin, *Introduction to Lattice Dynamics*. Cambridge, U.K.: Cambridge Univ. Press, 1983.
[13] 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.
[14] L. Guo and S.A. Billings, "State-space reconstruction and spatio-temporal prediction of lattice dynamical systems," *IEEE Trans. Autom. Control*, vol. 52, no. 4, pp. 622–632, Apr. 2007.
[15] L. Guo, S. A. Billings, and D. Coca, "Identification of partial differential equation models for a class of multiscale spatio-temporal dynamical systems," *Int. J. Control*, vol. 83, no. 1, pp. 40–48, 2010.

[16] H. Wei, S. A. Billings, Y. Zhao, and L. Guo, "Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio–temporal system identification," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 181–185, Jan. 2009.
[17] S. A. Billings, L. Guo, and H. Wei, "Identification of coupled map lattice models for spatio-temporal patterns using wavelets," *Int. J. Syst. Sci.*, vol. 37, no. 14, pp. 1021–1038, Nov. 2006.
[18] M. Milanese and C. Novara, "Set membership identification of nonlinear systems," *Automatica*, vol. 40, no. 6, pp. 957–975, 2004.
[19] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.
[20] S. A. Billings, S. Chen, and M. J. Korenberg, "Identification of MIMO non-linear systems using a forward-regression orthogonal estimator," *Int. J. Control*, vol. 49, no. 6, pp. 2157–2189, 1989.
[21] I. Lind and L. Ljung, "Regressor selection with the analysis of variance method," *Automatica*, vol. 41, no. 4, pp. 693–700, Apr. 2005.
[22] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press, 2002.
[23] D. Conus and R. C. Dalan, "The non-linear stochastic wave equation in high dimensions," *Electron. J. Probab.*, vol. 13, pp. 629–670, Jan. 2008
[24] J. A. K. Suykens, J. Vandewalle, and B. D. Moor, "Optimal control by least squares support vector machines," *Neural Netw.*, vol. 14, no. 1, pp. 23–35, Jan. 2001.
[25] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[26] J. Zhang, T. Sato, and S. Iai, "Novel support vector regression for structural system identification," *Struct. Control Health Monit.*, vol. 14, no. 4, pp. 609–626, Jun. 2007.
[27] S. Totterman, H. T. Toivonen, and B. Akesson, "Identification of state-dependent parameter models with support vector regression," *Int. J. Control*, vol. 80, no. 9, pp. 1454–1470, 2007.
[28] W. C. Chan, C. W. Chan, K. C. Cheung, and C. J. Harris, "On the modelling of nonlinear dynamic systems using support vector neural networks," *Eng. Appl. Artif. Intell.*, vol. 14, no. 2, pp. 105–113, 2001.
[29] Y. Liu, H. Wang, J. Yu, and P. Li, "Selective recursive kernel learning for online identification of nonlinear systems with NARX form," *J. Process Control*, vol. 20, no. 2, pp. 181–194, Feb. 2010.
[30] B. J. D. Kruif and T. J. A. D. Vries, "Pruning error minimization in least squares support vector machines," *IEEE Trans. Neural Netw.*, vol. 14, no. 3, pp. 696–702, May 2003.
[31] L. Guo, S. Mei, and B. A. Billings, "Neighbourhood detection and identification of spatio-temporal dynamical systems using a coarse-to-fine approach," *Int. J. Syst. Sci.*, vol. 38, no. 1, pp. 1–15, Jan. 2007.
[32] H. Wei and B. A. Billings, "Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information," *Int. J. Model. Identif. Control*, vol. 3, no. 4, pp. 341–356, 2008.
[33] M. Espinoza, J. A. K. Suykens, and B. D. Moor, "Kernel based partially linear models and nonlinear identification," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1602–1606, Oct. 2005.
[34] J. A. K. Suykens, V. T. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.
[35] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
[36] R. Reed, "Pruning algorithms-a survey," *IEEE Trans. Neural Netw.*, vol. 4, no. 5, pp. 740–747, Sep. 1993.
[37] H. Tang, S. Xue, R. Chen, and T. Sato, "Online weighted LS-SVM for hysteretic structural system identification," *Eng. Struct.*, vol. 28, no. 12, pp. 1728–1735, Oct. 2006.
[38] J. A. K. Suykens, L. Lukas, B. D. Moor, and J. Vandewalle, "Weighted least squares support vector machines: Robustness and sparse approximation," *Neurocomputing*, vol. 48, nos. 1–4, pp. 85–105, Oct. 2002.
[39] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Trans. Signal Process.*, vol. 43, no. 7, pp. 1713–1715, Jul. 1995.
[40] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.
[41] W. Liu, P. P. Pokharel, and J. C. Principe, "The kernel least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543–554, Feb. 2008.
[42] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.
[43] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*. New York: Wiley, 2010.
[44] M. Gerdts, G. Greif, and H. J. Pesch, "Numerical optimal control of the wave equation: Optimal boundary control of a string to rest in finite time," *Math. Comput. Simul.*, vol. 79, no. 4, pp. 1020–1032, Dec. 2008.
[45] C. Crossmann and H. G. Roos, *Numerical Treatment of Partial Differential Equations*. Berlin, Germany: Springer-Verlag, 2007.
[46] R. A. Carmona and B. Rozovskii, *Stochastic Partial Differential Equations: Six Perspectives*. Providence, RI: AMS, 1998.
[47] H. G. Matthies and A. Keese, "Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations," *Comput. Methods Appl. Mech. Eng.*, vol. 194, nos. 12–16, pp. 1295–1331, Apr. 2005.
[48] E. J. Allen, S. J. Novosel, and Z. Zhang, "Finite element and difference approximation of some linear stochastic partial differential equations," *Stochast. Stochast. Rep.*, vol. 64, nos. 1–2, pp. 117–142, 1998.
[49] H. Yoo, "Semi-discretization of stochastic partial differential equations on $\mathbb{R}_1$ by a finite-difference method," *Math. Comput.*, vol. 69, no. 230, pp. 653–666, Apr. 1999.

Hanwen Ning received the B.S. degree in applied mathematics and the Ph.D. degree in probability and mathematical statistics, both from the Huazhong University of Science and Technology, Wuhan, China, in 2005 and 2010, respectively.

He has been a Research Assistant in the Department of Mechanical Engineering, Hong Kong Polytechnic University, Hong Kong, since July 2010. His current research interests include identification, predictive control, numerical optimal control of nonlinear distributed parameter systems, control theory, and finite element methods of stochastic partial differential equation systems.

Xingjian Jing received the B.S. degree from Zhejiang University, Hangzhou, China, in 1998, the M.S. degree from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2001, and the Ph.D. degree in nonlinear systems and signal processing from the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K., in 2008.

He is currently an Assistant Professor in the Department of Mechanical Engineering, Hong Kong Polytechnic University (PolyU), Hong Kong. Before joining PolyU, he was a Research Fellow with the Institute of Sound and Vibration Research, University of Southampton, Southampton, U.K., from August 2008 to November 2009, where he worked on biological signal processing in collaboration with neuroscientists and funded by the Biotechnology and Biological Sciences Research Council (U.K.). He has published more than 50 papers in refereed journals and conference proceedings. His current research interests include system identification, signal processing, control of complex nonlinear systems, nonlinear analysis in the frequency domain, intelligent computing methods and their applications in nonlinear mechanical systems (sound and vibration control), nonlinear physiological systems (neural systems), robotic systems, and others.

Li Cheng received the B.S. degree in applied mechanics from Xi'an Jiaotong University, Xi'an, China, in 1984, and the DEA and Ph.D. degrees from the Institut National des Sciences Appliquees de Lyon, Lyon, France, in 1986 and 1989, respectively.

He joined the Laval University, Sillery, Canada, as an Assistant Professor in the Department of Mechanical Engineering in 1992. In the following years, he was promoted to Associate Professor and Full Professor. He joined the Department of Mechanical Engineering, Hong Kong Polytechnic University, Hong Kong, in 2000. He is currently a Chair Professor and the Director of the Consortium for Sound and Vibration Research. His current research interests include noise and vibration control, fluid-structure interaction, damage detection, and smart materials/structures.

Prof. Cheng has been a member of the Board of Directors and committees of different learning organizations such as the Acoustical Society of America, the Canadian Acoustical Society, the Acoustical Society of China, and the Chinese Society of Noise and Vibration Engineering. He also serves on the editorial boards or advisory committees of several international journals.