A spatio-temporal Volterra modeling approach for a class of distributed industrial processes
Han-Xiong Li, a,b,*, Chenkun Qi, a, Yongguang Yu, a,c

a Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong, PR China
b School of Mechanical and Electrical Engineering, Central South University, Changsha 410083, PR China
c Department of Mathematics, Beijing Jiaotong University, Beijing 100044, PR China

A R T I C L E   I N F O
Article history:
Received 23 June 2008
Received in revised form 20 February 2009
Accepted 28 February 2009

Keywords:
Distributed parameter system
System identification
Spatio-temporal Volterra
Galerkin method
Karhunen–Loève method
Laguerre polynomial

A B S T R A C T
It is difficult to model a distributed parameter system (DPS) due to the infinite-dimensional time/space nature and unknown nonlinear uncertainties. A low-dimensional and simple nonlinear model is often required for practical applications. In this paper, a spatio-temporal Volterra model is proposed with a series of spatio-temporal kernels for modeling unknown nonlinear DPS. To estimate these kernels, they are expanded onto spatial and temporal bases with unknown coefficients. To reduce the model dimension and parametric complexity in the spatial domain, the Karhunen–Loève (KL) method is used to find the dominant spatial bases. To reduce the parametric complexity in the temporal domain, the Laguerre polynomials are selected as temporal bases. Next, using the Galerkin method, this spatio-temporal modeling becomes a linear regression problem. Then unknown parameters can be easily estimated using the least-squares method in the temporal domain. After the time/space synthesis, the spatio-temporal Volterra model can be constructed. The convergence of parameter estimation can be guaranteed under certain conditions. This model has a low-dimensional and simple nonlinear structure, which is useful for the prediction and control of the DPS. The simulation and experiment demonstrate the effectiveness of the proposed modeling method.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Many physical and chemical industrial processes (e.g. fluid flow, thermal process and convection–diffusion–reaction process) are nonlinear distributed parameter systems (DPS) because their input, output, state and even parameters may vary both temporally and spatially. For the purpose of system simulation, dynamic prediction and control design, a proper model of the system is often needed. The first-principle modeling typically leads to various partial differential equations (PDE). While such models can accurately predict nonlinear and distributed dynamic behavior, their infinite-dimensional nature does not allow their direct use because of the limited actuators/sensors and computing power. In practice, a common approach is to approximate these systems using finite-dimensional models. Some model reduction approaches have been studied.

• Traditional time/space discretization methods, such as finite difference for the model reduction, can be easily applied to transform the PDE to the ordinary differential equation (ODE). However, they often lead to a high-dimensional approximate ODE, which is not very suitable for synthesizing implemental controller if the ODE is not further reduced.
• Advanced model reduction approaches are based on time/space separation using spatial basis function expansion. The idea comes from Fourier series expansion. A spatio-temporal variable can be expanded onto an infinite number of spatial basis functions with temporal coefficients. After choosing the proper finite number of spatial basis functions, a finite-dimensional model about temporal coefficients can be obtained by minimizing the approximation error (i.e., residual). This is so called the weighted residual method (WRM) [39]. The modeling accuracy and efficiency is very dependent on the choice of basis functions. Local basis function expansion in finite-element method will also lead to a high-dimensional ODE model. Fortunately global basis function expansion in spectral method [8] can derive an accurate low-dimensional ODE model, which is suitable for real-time control. The common used global basis functions include Fourier series [8], orthogonal polynomials [41], eigenfunctions of the system [39, 3, 11], and data based Karhunen–Loève (KL) empirical eigenfunctions [44, 36, 30, 2], whose choice depends on the type of PDE and boundary conditions. The model dimension can be further reduced using nonlinear WRM such as the inertial manifold method [12].
However, all these model reduction methods require that the PDE description of the system is accurately known. In many cases, it is difficult to obtain an exact PDE description of the industrial process and the data based identification must be employed. If the PDE structure of the system is known and only some parameters are unknown, then these parameters can be estimated from the process data (e.g. [18,4,13]). If some nonlinear terms of the PDE are unknown, then these nonlinearities can be approximately identified using intelligent modeling methods [19].

For completely unknown processes which widely exist in the industry, the black-box identification has to be used. System identification has been extensively studied for lumped parameter systems. However, only a few studies have been reported for distributed parameter systems. The main difficulty comes from the time/space coupling and complex nonlinearity of distributed parameter systems. This study will focus on the spatio-temporal modeling problem for a class of unknown nonlinear distributed parameter systems. Currently some black-box modeling approaches have been proposed.

- The PDE model identification methods have been studied in [26,25,6]. After recovering the PDE system from the spatio-temporal data, the model reduction methods are still needed for the practical applications.
- The time/space discretization of a DPS using finite difference will lead to a difference equation (also called lattice dynamical system). The identification of unknown lattice dynamical system has been studied in [38,27,14,7]. However the model may be of a high-dimension.
- The time/space separation of a DPS using spatial basis function expansion will lead to some ordinary differential equations. After choosing the proper local/global basis functions, the corresponding unknown ODE can be estimated using traditional system identification techniques. Some identification methods such as nonlinear autoregressive with exogenous input (NARX) model with local finite element basis functions [15], and neural networks with local Karhunen–Loève basis functions [55,45,1] have been studied. The use of local basis functions could lead to high-dimensional models. Though the modeling using global basis functions may result in low-dimensional models, however most of them use neural networks for temporal dynamics modeling which could have complex nonlinear structure.

Most of these modeling approaches for unknown DPS have some limitations for control, e.g. high-dimensional or complex nonlinear structure. Thus the development of low-dimensional and simple nonlinear model is necessary for control of unknown nonlinear DPS.

In general, the linear DPS can be represented by the impulse response function (i.e., Green's function and kernel). In some cases, the Green's function can be derived from the first-principle knowledge [10]. On the other hand, when the analytical Green's function is not available, it can be estimated from the input–output data [23,52–54,21]. However, the Green's function model uses one single kernel, which can only approximate the nonlinear system around the given working condition.

In the modeling of traditional lumped parameter systems, the fading memory nonlinear system (FMNS) [9] has been proposed to cover a wide range of industrial processes. Fading memory refers to the property that the dependence of system states on past inputs decreases rapidly with time. It means that the input signals far in the past have almost no effect on the present state of the system. The dynamics of the FMNS can be modeled by the Volterra series to any desired accuracy [9], where Volterra series are constructed by a series of kernels, from the 1st-order, the 2nd-order, and up to the high-order kernels. Because of its significant modeling capability, restricted model complexity and special nonlinear structure, extensive research has been reported for lumped system identification and control design using the Volterra model [42,40,22,34,37]. However, until now there has been almost no research reported for the application of the Volterra model to the DPS, because the traditional Volterra series do not have spatio-temporal nature. Though an infinite-dimensional Volterra series [5] can be used to model infinite-dimensional system, it is still a temporal series.

In order to model unknown nonlinear distributed parameter systems, the spatio-temporal kernel-based idea from the Green's function will be expanded into the Volterra series. A spatio-temporal Volterra model is constructed for the first time in this study by adding the space variables into the traditional Volterra model. This spatio-temporal Volterra model should be able to efficiently model a class of nonlinear DPS with a fading memory. In practice, many distributed parameter processes have fading memory features. Intuitively, for a DPS with a fading memory, two input signals which are close in the recent past, but not necessarily close in the remote past will yield very close current outputs. This also means the input in the remote fast has less influence to the current output. For an ODE system, there is no systematic or complete method to give the conditions of the fading memory property. In some cases, the conditions could arise. For example, a linear time-invariant (LTI) causal system has a convolution representation, or the internal dynamics of the system has a unique asymptotic state [9]. For a PDE system, it will be more difficult to check the fading memory property. This will need an investigation in the future. It could be inferred that a PDE that has a convolution representation (i.e., Green's function) may have a fading memory property. See Refs. [23,10] for some examples.

The spatio-temporal Volterra modeling approach is designed as follows. A spatio-temporal Volterra model is proposed first with a set of spatio-temporal kernels for modeling unknown nonlinear DPS. Then in order to estimate the spatio-temporal kernels from the input–output data, each kernel is time/space separated with the expansion onto spatial and temporal basis functions with unknown coefficients. To reduce the model dimension and parametric complexity in the spatial domain, the KL method is used to find the dominant spatial output basis functions. To reduce the parametric complexity in the temporal domain, the Laguerre polynomials are selected as the temporal basis functions. Next, using the Galerkin method, this spatio-temporal modeling problem will turn into a temporal modeling problem. Then unknown parameters can be easily estimated using the least-squares method in the temporal domain. After the time/space synthesis of kernels, the spatio-temporal Volterra model can be constructed. The convergence of parameter estimation can be guaranteed under certain conditions. The state space representation of spatio-temporal Volterra model can be easily obtained in a form of blocked-oriented nonlinear Wiener structure. The simulation and the experiment demonstrate the effectiveness of the proposed modeling method.

This spatio-temporal Volterra model is useful for modeling a class of nonlinear DPS with a fading memory. The selection of temporal bases and spatial output bases also influences what kind of DPS can be modeled. If the time-scaling factor is chosen suitably, then the Laguerre functions can efficiently model stable lumped systems [50]. On the other hand, KL basis functions are efficient for most of parabolic PDE, where the eigenspectrum can be separated into slow and fast modes [2]. Therefore, when using Laguerre and KL basis functions, the proposed modeling approach can efficiently model stable parabolic PDE with a fading memory. The estimated spatio-temporal Volterra model has the advantage of low-dimension and simple nonlinear structure, which is desired for control of nonlinear DPS. Using the KL spatial basis functions, the spatio-temporal Volterra model can be low-dimen-
sional. This model is also a complexity restricted model with a simple nonlinear structure since it is a spatial extension of traditional Volterra series. Thus the identification and control approaches for the traditional Volterra model can be extended to DPS. Moreover, the spatio-temporal Volterra model can be transformed into a spatio-temporal Wiener form which has a blocked-oriented nonlinear structure, so control methods for traditional Wiener system can be easily extended to DPS. It is also easy to apply popular linear control to nonlinear DPS since this model is a nonlinear extension of linear Green’s function.

This paper is organized as follows. The spatio-temporal Volterra model is proposed in Section 2. Section 3 presents the spatio-temporal modeling approach. The state space realization is provided in Section 4. Section 5 gives the parameter convergence analysis. The simulation and the experiment are demonstrated in Section 6.

2. Spatio-temporal Volterra model

It is well known that a linear continuous DPS can be represented as a linear mapping from the input $u(x,t)$ to the output $y(x,t)$, where $x \in \Omega$ denotes space variable, $\Omega$ is the spatial domain, and $t$ is time. This mapping can be expressed in a Fredholm integral equation of the first kind containing a square-integrable kernel $g$ (i.e., impulse response function or Green’s function) [10,23]

$$y(x,t) = \int_{\Omega} \int_{0}^{t} g(x, t', t) u(x, t') dt' dt.$$  

(1)

On the other hand, a lumped parameter system (LPS) $y(t) = N(u(t)) + d(t)$, where $u(t)$ is the input, $t$ is the discretized time instant, $y$ and $d$ are the output and the stochastic disturbance, and $N$ is an operator with a fading memory, can be approximated by a discrete-time Volterra model [9]

$$y(t) = \sum_{r=1}^{n} \sum_{\tau_1=0}^{m_{r}} \cdots \sum_{\tau_{r}=0}^{m_{r}} g_r(t, \tau_1, \ldots, \tau_r) \prod_{j=1}^{r} u(\tau_j),$$  

(2)

where $g_r$ is the $r$-th order temporal kernel.

Motivated by (1) and (2), for a distributed parameter system (DPS) $y(x,t) = N(u(x,t)) + d(x,t)$, where $u(x,t)$ is $u(t)$, the input, and $x$ is the spatial variable, the spatio-temporal Volterra model is constructed by adding the space variables into the traditional Volterra model

$$y(x,t) = \sum_{r=1}^{\infty} \int_{\Omega} \cdots \int_{\Omega} \cdots \int_{\Omega} g_r(x, t', \ldots, t_r) \prod_{j=1}^{r} u(\tau_j) \prod_{j=1}^{r} d(\tau_j),$$  

(4)

Similarly the model (4) can also work for the space-varying or time-invariant system. When the model is homogeneous in the spatial domain, there exists

$$g_r(x, s_1, \ldots, s_r, t, \tau_1, \ldots, \tau_r) = g_r(x, s_1, \ldots, s_r, t - \tau_1, \ldots, t - \tau_r).$$  

(5)

$$g_r(x, s_1, \ldots, s_r, t, \tau_1, \ldots, \tau_r) = g_r(x, s_1, \ldots, s_r, t - \tau_1, \ldots, t - \tau_r).$$  

(6)

In this study, we only consider the time-invariant and space-varying case (5) since it is very common in the real applications. Substituting (5) into (4) will have the following expression:

$$y(x,t) = \sum_{r=1}^{\infty} \int_{\Omega} \cdots \int_{\Omega} \cdots \int_{\Omega} g_r(x, \ldots, t_r) \prod_{j=1}^{r} u(\tau_j, t - \tau_j) d(\tau_j),$$  

(7)

Note that an intuitive derivation of traditional Volterra series for LPS using Taylor expansion is given in [40]. This intuitive approach can be extended to DPS (see Appendix A). Though the derivation of Volterra series for DPS using Taylor expansion may not be rigorous, it can help to understand the working principle of spatio-temporal Volterra series (7) proposed here. The model (7) is still not applicable because of its infinite-order. In practice, the higher order terms can be neglected and only the first $k$ kernels need to be taken into account as below

$$y(x,t) = \sum_{r=1}^{k} \int_{\Omega} \cdots \int_{\Omega} \cdots \int_{\Omega} g_r(x, \ldots, t_r) \prod_{j=1}^{r} u(\tau_j, t - \tau_j) d(\tau_j),$$  

(8)

where the error term $v(x,t)$ includes unmodeled dynamics and external noise. The modeling accuracy and the model complexity can be controlled by the order $R$.

3. Spatio-temporal modeling approach

Now the problem is to estimate a spatio-temporal Volterra model (8) from a set of spatio-temporal input–output data $\{u(x, t) | x \in \Omega, t = 1, \ldots, n_i\}, \{y(x, t) | x \in \Omega, J = 1, \ldots, n_j, t = 1, \ldots, n_t\}$, where $n_i$ denotes the time length and $n_j$ is the number of sampled spatial points of the output. For simplicity, it is assumed that the spatial information of the input is known from some physical knowledge and the locations $x_j | j = 1, \ldots, n_j$ are uniformly distributed over the spatial domain. In order to achieve a good modeling performance, the order $R$ can be determined in an incremental way using the cross-validation technique or some optimization methods. Once the order is determined, the next problem is to estimate the kernels. The main difficulty comes from the time/space coupling of kernels.

Using a simple time/pace discretization for kernels $g_r(x, s_1, \ldots, s_r, t, \tau_1, \ldots, \tau_r)$ ($r = 1, \ldots, R$) will lead to a large amount of parameters to be estimated. However, it is important to reduce the model dimension and parametric complexity, improve the numerical condition and decrease the variance of the estimated parameters. This can be done using the time/space method, i.e., expanding the kernels in terms of a relatively small number of orthonormal basis functions such as KL spatial bases and Laguerre temporal bases. After the time/space separation, the original spatio-temporal problem will turn to the traditional temporal modeling problem. Thus, the unknown parameters can be easily estimated in the temporal domain. Finally, the spatio-temporal Volterra model can be reconstructed using the time/space synthesis. The modeling idea is shown in Fig. 1. The time/space separation and synthesis are very critical for this identification approach, which are the key differences from the traditional Volterra modeling.
3.1. Time/space separation

For simplicity, the input $u(x, t)$ is assumed to have a finite-dimensional freedom since only a finite number of actuators are available in practice. Therefore the input $u(x, t)$ can be formulated in terms of a finite number of orthonormal spatial input basis functions $\{\psi_i(x)\}_{i=1}^m$:

$$u(x, t) = \sum_{i=1}^m \psi_i(x) a_i(t),$$

(9)

where $a_i(t) = \int u(x, t) \psi_i(x) dx$ is the time coefficient (input signal) and $m$ is the number of actuators. Ideally, the output $y(x, t)$ and the error $v(x, t)$ can be expressed by an infinite set of orthonormal spatial output basis functions $\{\theta_i(x)\}_{i=1}^\infty$:

$$y(x, t) = \sum_{i=1}^\infty \phi_i(x) b_i(t),$$

(10)

$$v(x, t) = \sum_{i=1}^\infty \psi_i(x) v_i(t),$$

(11)

where $b_i(t) = \int y(x, t) \phi_i(x) dx$ and $v_i(t) = \int v(x, t) \phi_i(x) dx$ are the time coefficients of the output and error respectively. This is because of the inherently infinite-dimensional characteristic of the DPS. Practically, for most of parabolic systems, both output $y(x, t)$ and error $v(x, t)$ can be truncated into $n$ dimensions as below:

$$y_n(x, t) = \sum_{i=1}^n \phi_i(x) b_i(t),$$

(12)

$$v_n(x, t) = \sum_{i=1}^n \psi_i(x) v_i(t).$$

(13)

The dimension $n$ will be dependent on how the eigenspectrum of the DPS is separated into slow and fast modes, the type of spatial basis functions and the required modeling accuracy. For convenience, define $\mathbf{a}(t) = [a_1(t), \ldots, a_m(t)]^T \in \mathbb{R}^m$, $\mathbf{b}(t) = [b_1(t), \ldots, b_n(t)]^T \in \mathbb{R}^n$, and $\mathbf{u}(t) = [u_1(t), \ldots, u_n(t)]^T \in \mathbb{R}^n$.

Assuming that the kernels in (8) are absolutely integrable on the temporal domain $[0, \infty)$ at any spatial point $x$ and $\zeta$, which means that the corresponding spatio-temporal Volterra model is stable, then they can be represented by means of orthonormal temporal basis functions. The kernels are supposed to be expanded onto input bases $\{\psi_i(x)\}_{i=1}^m$, output bases $\{\phi_i(x)\}_{i=1}^n$, and temporal bases $\{\phi_i(t)\}_{i=1}^q$:

$$g_r(\cdot) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k_1=1}^q \cdots \sum_{k_r=1}^q \theta_{ij_1 \cdots j_r k_1 \cdots k_r} \phi_i(x) \prod_{p=1}^r \psi_j(\zeta_p) \phi_{k_p}(\tau_{p}),$$

(14)

where $\theta_{ij_1 \cdots j_r k_1 \cdots k_r}$ is the corresponding constant coefficient of the $r$th-order kernel onto output bases $\phi_i(x)$, input bases $\psi_j(\zeta_p)$, and temporal bases $\phi_{k_p}(\tau_{p})$.

3.2. Temporal modeling problem

Substitution of (9) and (12)--(14) into (8) will have:

$$\sum_{h=1}^n \phi_h(x) b_h(t) = \sum_{r=1}^N \int \int \cdots \int \prod_{i=1}^r d\tau_i \prod_{j=1}^q d\zeta_j \int \prod_{i=1}^m \psi_i(\zeta_i) \phi_i(x) dx \prod_{p=1}^r \psi_j(\zeta_p) \phi_{k_p}(\tau_{p}) \phi_h(x) v_i(t)$$

(15)

$$+ \sum_{h=1}^n \phi_h(x) v_h(t).$$

Eq. (15) can be further simplified into:

$$\sum_{h=1}^n \phi_h(x) b_h(t) = \sum_{r=1}^R \sum_{j_1=1}^m \sum_{j_2=1}^m \cdots \sum_{k_1=1}^q \cdots \sum_{k_r=1}^q \theta_{ij_1 \cdots j_r k_1 \cdots k_r}$$

$$\times \prod_{i=1}^m \psi_i(\zeta_i) \phi_i(x) \prod_{j=1}^m \psi_j(\zeta_j) \phi_j(x) \prod_{p=1}^r \psi_{j_p}(\zeta_{p}) \phi_{k_p}(\tau_{p})$$

$$\times \sum_{h=1}^n \phi_h(x) v_h(t),$$

(16)

where \(\psi_{j_p}(\cdot) = \int \psi_{j_p}(\zeta) \psi_{w_{j_p}}(\zeta) d\zeta\),

\(l_{k_p}(t) = \sum_{\tau=0}^t \phi_k(\tau) a_k(t - \tau).\)

Using the Galerkin method [11], projecting (16) onto the output basis functions $\phi_h(x)(h = 1, \ldots, n)$ will lead to the following expression:
In practice, 

\[ H = \frac{1}{2} \]

where \( b = 3.3 \).

### 3.3. Parameter identification

Re-arranging the order of integration and summation, it becomes

\[
\sum_{h=1}^{n} \varphi_{h} b_{2} = \sum_{r=1}^{m} \sum_{k=1}^{q} \sum_{l=1}^{p} \sum_{n=1}^{t} \varphi_{r} \varphi_{l} \varphi_{n} \varphi_{t} \times \sum_{u=x}^{w} \sum_{v=x}^{w} \sum_{m=1}^{M} \sum_{n=1}^{N} \varphi_{h} \varphi_{t} (t),
\]

where \( \varphi_{h} = \int_{\Omega} \varphi_{h} \varphi_{t} \).

Since the matrix \( \varphi_{h} \) is invertible due to the orthonormal bases, the following expression can be derived from (19).

\[
b(t) = \sum_{r=1}^{m} \sum_{k=1}^{q} \sum_{l=1}^{p} \sum_{n=1}^{t} \varphi_{r} \varphi_{l} \varphi_{n} \varphi_{t} \times \sum_{u=x}^{w} \sum_{v=x}^{w} \sum_{m=1}^{M} \sum_{n=1}^{N} \varphi_{h} \varphi_{t} (t) + u(t).
\]

where

\[
\varphi_{r} \varphi_{l} \varphi_{n} \varphi_{t} = \left[ \varphi_{r} \varphi_{l} \varphi_{n} \varphi_{t} \right]^{T} \in \mathbb{R}^{n}.
\]

### 3.3. Parameter identification

Eq. (20) can be expressed in a linear regression form

\[
b(t) = \Theta^{T} \Phi(t) + u(t),
\]

where

\[
\Theta = [\Theta^{1}, \Theta^{2}, \ldots, \Theta^{R}]^{T} \in \mathbb{R}^{(mpq \times m^{2}) \times n},
\]

\[
\Phi = [\Phi^{1}, \Phi^{2}, \ldots, \Phi^{R}]^{T} \in \mathbb{R}^{m \times m^{2}},
\]

and

\[
\Theta^{(r)} = \left[ \Theta_{r,1}, \Theta_{r,2}, \ldots, \Theta_{r,mpq} \right] \in \mathbb{R}^{n \times m^{2}},
\]

\[
\Phi^{(r)} = \left[ \Phi_{r,1}, \Phi_{r,2}, \ldots, \Phi_{r,mpq} \right] \in \mathbb{R}^{m \times m^{2}},
\]

\[
\Theta^{(r)} = \left[ \Theta_{r,1}, \Theta_{r,2}, \ldots, \Theta_{r,mpq} \right] \in \mathbb{R}^{n \times m^{2}},
\]

In practice, \( u \) and \( y \) are uniformly sampled over the spatial domain. In this case, \( a \) and \( b \) can also be computed from the pointwise data using spline interpolation in the spatial domain. Then, \( \Phi(t) \) can be constructed from \( a \).

Now considering the \( n \) set of temporal data \( \{a(t)\}_{t=1}^{n}, \{b(t)\}_{t=1}^{n} \), it is well known from [33] that by minimizing a quadratic criterion of the prediction errors

\[
\hat{\Theta} = \arg \min_{\Theta} \left\{ \frac{1}{n} \sum_{t=1}^{n} \left[ b(t) - \Theta^{T} \Phi(t) \right]^{2} \right\},
\]

where \( \|b(t)\|^{2} = \hat{r}^{T} \Phi(t) \hat{r} \), \( \Theta \) can be estimated using the least-squares method as follows:

\[
\hat{\Theta} = \left[ \frac{1}{n} \sum_{t=1}^{n} \Phi(t) \Phi^{T}(t) \right]^{-1} \left[ \frac{1}{n} \sum_{t=1}^{n} \Phi(t) b^{T}(t) \right],
\]

provided that the indicated inverse exists. This condition can be guaranteed when using persistently exciting input.

After the kernels in (14) are reconstructed using the time/space synthesis from the estimated parameters \( \Theta \), the spatio-temporal Volterra model can be obtained from (8).

### 4. State space realization

The spatio-temporal Volterra model (8) can also be transformed into a state space form. The Laguerre network representation of the realization procedure is shown in Fig. 2, where the transfer functions

\[
G_{1}(s) = \frac{\sqrt{2p}}{s + p}, \quad G_{2}(s) = \cdots = G_{w}(s) = \frac{s - p}{s + p},
\]

can be derived from (47) in Appendix C, and \( p \) is the time-scaling factor.

The variable \( l_{0w}(t) \) in (18) is defined as the state. It can be shown that the state satisfies the following difference equations [51]

\[
L_{w}(t + 1) = K_{w} L_{w}(t) + H_{w} a_{w}(t), \quad w = 1, \ldots, m,
\]

where \( L_{w}(t) = [l_{1w}, \ldots, l_{mw}]^{T} \in \mathbb{R}^{m}, \) the matrices \( K_{w} \in \mathbb{R}^{mq \times q} \) and \( H_{w} \in \mathbb{R}^{q} \) are defined in Appendix D.

Finally, the state equation can be written as

\[
L(t + 1) = K(t) L(t) + H a(t),
\]

where \( L(t) = [L_{1}, \ldots, L_{m}]^{T} \in \mathbb{R}^{m}, \) \( K = \text{diag}(K_{w}) \in \mathbb{R}^{mq \times m} \) and \( H = \text{diag}(H_{w}) \in \mathbb{R}^{m \times m}. \) The output equation can be derived from (16) as follows:

\[
y_{w}(x, t) = C(x)^{T} L(t) + D(x) L(t) + \cdots,
\]

where the spatial matrices \( C(x) \) and \( D(x) \) are given in Appendix D.
It can be seen from (27) and (28) that the spatio-temporal Volterra model can be transformed into a spatio-temporal Wiener model in a state space form (a linear lumped parameter system followed by a static spatio-temporal nonlinearity). With this Wiener model, traditional lumped control design with Laguerre model, such as model predictive control [49] and adaptive control [51], can be extended to nonlinear DPS. Moreover, many control methods for lumped Wiener model can also be extended to nonlinear DPS.

5. Parameter convergence analysis

For lumped parameter systems modeling, traditional Volterra series can approximate a class of nonlinear systems with a fading memory at any desired accuracy [9]. So the proposed spatio-temporal Volterra model should be able to efficiently model a class of nonlinear time-invariant DPS with a fading memory. In the following section, the convergence of the estimated parameters and model to their optimal values in the identification is proved. Here the optimal Volterra model is the best available Volterra model that we can get in the given model structure (finite-dimensional), which may not be very close to the original PDE. The convergence of spatio-temporal Volterra series to original PDE is still an open problem, which is worth an investigation in the future.

For simplicity, let $y_n(x,t) = V(x,t, \Theta, \{u(\zeta, \tau)\})$ denote a finite-order spatio-temporal Volterra model with $R, n, q < \infty$

$$y_n(x,t) = \sum_{i=1}^{R} \int_{\Omega} \cdots \int_{\Omega} x_{i-1}^{m} \cdots x_{0}^{m} \cdots \sum_{j=1}^{q} \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \phi_j(x) \phi_{i,j} \phi_{i,j,k_1,k_2} \delta(x-x_i, t-t_i) dx_i,$$

where the parameter matrix $\Theta$ is defined by (23).

Because the functions $\phi_i(x) \phi_{i,j} \phi_{i,j,k_1,k_2}$ are basis functions, the spatio-temporal Volterra model structure $y_n(x,t) = V(x,t, \Theta, \{u(\zeta, \tau)\})$ is identifiable, i.e.,

$$V(x,t, \Theta_1, \{u(\zeta, \tau)\}) = V(x,t, \Theta_2, \{u(\zeta, \tau)\}) \Rightarrow \Theta_1 = \Theta_2.$$

For a nonlinear time-invariant DPS $y(x,t) = N(u(\zeta, \tau)) + d(x,t)$ with a fading memory on the input set $U$, and a finite-order spatio-temporal Volterra model $y_n(x,t) = V(x,t, \Theta, \{u(\zeta, \tau)\})$ with $R, n, q < \infty$, we always assume that there exists a nonempty parameter set $D_\Theta = \{\Theta, \Omega \in \Theta | y_n(x,t) \neq V(x,t, \Theta, \{u(\zeta, \tau)\})\}$.

Then there is an optimal model

$$y_n(x,t) = V(x,t, \Theta^*, \{u(\zeta, \tau)\}),$$

with an optimal parameter matrix $\Theta^*$ such that

$$\Theta^* = \arg \min_{\Theta \in D_\Theta} \{\|E(y_n(x,t) - V(x,t, \Theta, \{u(\zeta, \tau)\})\|^2\} \equiv \{\Theta | y_n(x,t) = V(x,t, \Theta, \{u(\zeta, \tau)\})\} \equiv \delta, \forall u \in U, \forall t \in T.$$

6. Case study

In order to evaluate the proposed modeling method, two typical examples are studied. The first one is a catalytic rod that is of one-dimensional space, and the second one is a catalytic plate that is of two-dimensional space. For an easy comparison, some performance indexes are established for the DPS as follows:

- Spatial normalized absolute error, $\text{SNAE}(t) = \int |\varepsilon(x,t)| dx / \| dx, $
- Temporal normalized absolute error, $\text{TNAE}(x) = \sum |\varepsilon(x,t)| / \sum \Delta t, $
- Root of mean squared error, $\text{RMSE} = (\sum \varepsilon(x,t)^2 / \| dx \sum \Delta t)^{1/2}.$

6.1. Catalytic rod

A long thin rod in a reactor as shown in Fig. 3 is a typical transport–reaction process in the chemical industry [11]. The reactor is fed with pure species $A$ and a zeroth order exothermic catalytic reaction of the form $A \rightarrow B$ takes place in the rod. Since the reaction is exothermic, a cooling medium that is in contact with the rod is used for cooling. Under the assumptions of constant density and heat capacity of the rod, constant conductivity of the rod, and constant temperature at both sides of the rod, and excess of species $A$ in the furnace, the mathematical model which describes the spatio-temporal evolution of the rod temperature consists of the following parabolic partial differential equation [11].
\[
\frac{\partial y(x,t)}{\partial t} = \frac{\partial^2 y(x,t)}{\partial x^2} + \beta_T \left( e^{-\gamma q} - e^{-q} \right) + \beta_m (\psi(x)^T a(t) - y(x,t)),
\]

subject to the Dirichlet boundary and initial conditions

\[
y(0, t) = 0, y(\pi, t) = 0,
y(x, 0) = y_0(x),
\]

where \(y(x,t), \psi(x), \beta_T, \beta_m, \gamma\) and \(a\) denote the temperature in the reactor, the actuator distribution, the heat of reaction, the heat transfer coefficient, the activation energy, and the manipulated input (temperature of the cooling medium). The process parameters are often set as \(\beta_T = 50, \beta_m = 2, \gamma = 4\).

There are available four actuators \(a(t) = [a_1(t), \ldots, a_4(t)]^T\) with the spatial distribution function \(\psi(x) = [\psi_1(x), \ldots, \psi_4(x)]^T, \psi_i(x) = H(x - (i-1)\pi/4) - H(x - i\pi/4), (i = 1, \ldots, 4)\) and \(H(\cdot)\) is the standard Heaviside function. Twenty-two sensors uniformly distributed in the space are used to measure the temperature distribution. The sampling interval is \(\Delta t = 0.01\).

In the simulation, the random input signals are used to excite this process. For example, all the 2500 samples for actuator 1 with the simulation time 25 are shown in Fig. 4, where the first 1800 samples are training input and the remaining 700 samples are testing input. Correspondingly a total of 2500 output measurements are collected, where the first 1800 data are used for model training and the last 700 data for model testing.

In the spatio-temporal Volterra modeling, the temporal bases \(\phi(t)\) are chosen as Laguerre functions with the time-scaling factor \(p = 4.05\) and the truncation length \(q = 4\). For the spatial output bases, as shown in Fig. 5 the first four Karhunen–Loève basis functions are used. It should be noted that in this study, we assume a basic situation, i.e., the model and the controlled system will work around the working conditions. In order to obtain the significant spatial basis functions which can work well around the working points, a basic requirement of KL method is that the data must be representative. It means the data should cover the characteristic dynamics around the working points. Then after the significant basis functions are obtained offline, it is not necessary to change them online when the model is used in control. If the controlled system works outside this range, it requires an adaptive model—a much more complex issue.

The first 250 output measurements \(y(x, t)\) in the training data set are shown in Fig. 6. The corresponding predicted output \(\hat{y}_n(x, t)\) and prediction error \(e(x, t) = y - \hat{y}_n\) of the 2nd-order...
spatio-temporal Volterra model are presented in Figs. 7 and 8 respectively. The model also achieves very good performance over the remaining data set. In addition, the measured and predicted output at sensor 11 (x = 10π/21) and sensor 16 (x = 15π/21) over the whole data set are shown in Figs. 9 and 10. The model also works very well at other sensor locations. It is obvious that the 2nd-order spatio-temporal Volterra model can satisfactorily model the process. For many applications, the 2nd-order model is enough for a good approximation, and too high-order models may cause the over-complexity problem. As shown in Figs. 11 and 12, the 2nd-order spatio-temporal Volterra model performs much better than the 1st-order spatio-temporal Volterra model (i.e., Green’s function model) because the 1st-order spatio-temporal Volterra model is basically a linear system. To better understand the application of this modeling method, the details of each stage in Fig. 1 for the 2nd-order spatio-temporal modeling are given in Appendix F.

Because traditional finite-element methods for known PDE are not suitable for unknown DPS, in order to provide a comparison with the modeling using finite element bases, a 2nd-order Volterra model is identified with four third-order splines as spatial output basis functions (see Fig. 13). See the references [15,43] for details on the construction of spline functions. The performance index SNAE(t) and TNAE(x) over the whole data set in Figs. 14 and 15 show that the modeling with KL basis functions is more accurate than the modeling with the same number of spline basis functions. This is because of the optimality of the KL basis functions.

![Fig. 7. Predicted output of 2nd-order Volterra model.](image1)

![Fig. 8. Prediction error of 2nd-order Volterra model.](image2)

![Fig. 9. Performance of 2nd-order Volterra model at sensor s11 (x = 10π/21).](image3)

![Fig. 10. Performance of 2nd-order Volterra model at sensor s16 (x = 15π/21).](image4)
The modeling performance is also affected by the number of Laguerre temporal basis functions (q) and the time-scaling factor (p). Fig. 16 displays RMSE of the 2nd-order spatio-temporal Volterra model with respect to these two parameters. It is shown that this modeling approach is robust to these parameters since there are a wide range of parameters which can be chosen to obtain a good performance.

6.2. Snap curing oven

In the semiconductor back-end packaging process, the snap curing oven (Fig. 17) is an important equipment to provide the required curing temperature distribution [19]. After epoxy die attach, the bonded leadframe to be cured are moved in and out from the inlet and outlet. For better control, a model for estimating the temperature distribution inside the chamber is required. As shown in Fig. 18, it is equipped with four heaters (h1–h4) and sixteen temperature sensors (s1–s16). In the experiment, the random input signals are used to excite this thermal process and the total 2100 samples for heater 1 are shown in Fig. 19. Correspondingly a total of 2100 output measurements are collected with a sampling interval $\Delta t = 10$ s. One thousand and four hundreds of measure-
ments from sensors (s1–s5, s7–s10, and s12–s16) are used to train the model. The last 700 measurements from sensors (s1–s5, s7–s10, and s12–s16) are chosen to evaluate the model for untrained data. All 2100 measurements from the rest sensors (s6, s11) are used for modeling performance verification.

In the spatio-temporal Volterra modeling, five two-dimensional Karhunen–Loève basis functions are used as spatial bases and the first two of them are shown in Figs. 20 and 21. The temporal bases \( \phi_i(t) \) are chosen as Laguerre series with the time-scaling factor \( p = 0.001 \) and the truncation length \( q = 3 \).

The 2nd-order spatio-temporal Volterra model is used to model the thermal process. After the training using the first 1400 data from the sensors (s1–s5, s7–s10, and s12–s16), a process model can be obtained with the significant performance such as the sensor s1 in Fig. 22. The model also performs very well for the untrained locations such as the sensor s6 in Fig. 23. The predicted temperature distribution of the oven at \( t = 10,000 \) s is provided in Fig. 24. The performance comparisons over the whole dataset in Figs. 25, 26 and Table 1 further show that the 2nd-order spatio-temporal Volterra model has a much better performance than the 1st-order spatio-temporal Volterra model. As shown in Fig. 27, a good modeling performance can be maintained in a wide range of parameter space about the number of Laguerre temporal basis functions (\( q \)) and the time-scaling factor (\( p \)).

To compare with the modeling using finite element bases, a 2nd-order spatio-temporal Volterra model is also constructed using nine third-order splines as spatial basis functions. The first two of them are shown in Figs. 28 and 29. The performance index SNAE(t) in Fig. 30 and TNAE(x) in Table 2 over the whole data set show that the modeling with KL basis functions is much better than the modeling even with more number of spline basis functions. It means that to achieve good accuracy, the modeling with KL basis functions could require a smaller number of parameters than the modeling with spline basis functions because the dimension of the model with KL basis functions could be lower (i.e., the number of KL basis functions used is less than the number of splines functions) due to the optimality of KL basis functions. The effectiveness of the proposed modeling method is clearly demonstrated in this real application.

7. Conclusion

The development of a low-dimensional and simple nonlinear model is necessary for unknown nonlinear DPS. In this paper, a spatio-temporal Volterra modeling approach is proposed for unknown nonlinear DPS. The Volterra kernels are functions of time and space variables. In order to reduce the model dimension and parametric complexity, the kernels are expanded onto Karhunen–Loève spatial bases as well as Laguerre temporal bases with un-
known coefficients. With the help of the Galerkin method, these unknown parameters can be estimated from the process data using the least-squares method in the temporal domain. The convergence of parameter estimation can be guaranteed under certain conditions. This spatio-temporal Volterra model can achieve a better performance than the Green’s function model. It is useful for the prediction and control of the nonlinear spatio-temporal field.
The simulation and experiment are conducted to demonstrate the effectiveness of the proposed modeling method and its potential application to a wide range of nonlinear DPS. The convergence of spatio-temporal Volterra series to original PDE is still an open problem, which is worth to investigate in the future.

Acknowledgements

Authors would like to thank the associate editor and the anonymous referees for their valuable comments and suggestions. The work is partially supported by a grant from RGC of Hong Kong (CityU: 117208) and a SRG grant from City University of Hong Kong (7002219) and a grant from NSF China (50775224).

Table 1

<table>
<thead>
<tr>
<th></th>
<th>1st-order model</th>
<th>2nd-order model</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>1.87</td>
<td>0.79</td>
</tr>
<tr>
<td>s2</td>
<td>1.44</td>
<td>0.71</td>
</tr>
<tr>
<td>s3</td>
<td>1.75</td>
<td>1.2</td>
</tr>
<tr>
<td>s4</td>
<td>1.34</td>
<td>0.71</td>
</tr>
<tr>
<td>s5</td>
<td>1.75</td>
<td>1.59</td>
</tr>
<tr>
<td>s6</td>
<td>1.55</td>
<td>0.65</td>
</tr>
<tr>
<td>s7</td>
<td>1.5</td>
<td>0.78</td>
</tr>
<tr>
<td>s8</td>
<td>1.59</td>
<td>1.24</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th></th>
<th>1st-order model</th>
<th>2nd-order model</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>1.13</td>
<td>0.86</td>
</tr>
<tr>
<td>s2</td>
<td>1.81</td>
<td>0.86</td>
</tr>
<tr>
<td>s3</td>
<td>2.1</td>
<td>1.33</td>
</tr>
<tr>
<td>s4</td>
<td>1.43</td>
<td>1.08</td>
</tr>
<tr>
<td>s5</td>
<td>1.85</td>
<td>0.71</td>
</tr>
<tr>
<td>s6</td>
<td>2.22</td>
<td>1</td>
</tr>
<tr>
<td>s7</td>
<td>1.13</td>
<td>0.78</td>
</tr>
<tr>
<td>s8</td>
<td>1.52</td>
<td>0.55</td>
</tr>
</tbody>
</table>

The simulation and experiment are conducted to demonstrate the effectiveness of the proposed modeling method and its potential application to a wide range of nonlinear DPS. The convergence of spatio-temporal Volterra series to original PDE is still an open problem, which is worth to investigate in the future.

Acknowledgements

Authors would like to thank the associate editor and the anonymous referees for their valuable comments and suggestions. The work is partially supported by a grant from RGC of Hong Kong (CityU: 117208) and a SRG grant from City University of Hong Kong (7002219) and a grant from NSF China (50775224).
Appendix A. Intuitive derivation of spatio-temporal Volterra series using Taylor expansion

It is natural to view the output $y(x, t)$ of a nonlinear distributed parameter system at a particular space $x$ and time $t$ as depending (in a nonlinear way) on $x$ and all values of the input at time prior to $t$ and space for all $\zeta \in \Omega$. That is, $y(x, t)$ depends on $x$ and $u(\zeta, t - \tau)$ for all $\zeta \in \Omega$ and $\tau \geq 0$.

If $u(\zeta, t - \tau)$ for all $\zeta \in \Omega$ and $\tau \geq 0$ can be characterized by a set of $u_i(t), u_j(\zeta)$, then the output $y(x, t)$ can be represented as a nonlinear function

$$y(x, t) = f(x, u_1(t), u_2(t), \ldots).$$

(33)

Suppose that $t$ is fixed and the input $u(\zeta, t - \tau), \zeta \in \Omega$, $0 \leq \tau < \infty$, is an element of the Hilbert space of square-integrable functions $L_2(\Omega \times [0, \infty])$. That is,

$$\int_0^\infty \int_0^\infty u^2(\zeta, t - \tau) d\zeta dt < \infty.$$  

(34)

Furthermore, suppose that $\psi_1(\zeta, \tau), \psi_2(\zeta, \tau), \ldots$ is an orthonormal basis for this space:

$$\int_0^\infty \psi_i(\zeta, \tau) \psi_j(\zeta, \tau) d\zeta dt = \begin{cases} 1, & i = j, \\ 0, & i \neq j \end{cases}$$

(35)

Then the value of the input signal at any space and any time in the past can be written in the form

$$u(\zeta, t - \tau) = \sum_{i=1}^\infty u_i(t) \psi_i(\zeta, \tau),$$

(36)

where

$$u_i(t) = \int_0^\infty \int_0^\infty u(\zeta, t - \tau) \psi_i(\zeta, \tau) d\zeta dt.$$  

(37)

Expand the function $y(x, t) = f(x, u_1(t), u_2(t), \ldots)$ into a power series so that the output at space $x$ and time $t$ is

$$y(x, t) = \sum_{i=1}^\infty a_i(x) u_i(t) + \sum_{i=1}^\infty \sum_{j=1}^\infty a_{ij}(x) u_i(t) u_j(t) + \cdots$$

(38)

Substituting (37) into (38) will obtain

$$y(x, t) = \int_0^\infty \int_0^\infty \sum_{i=1}^\infty a_i(x) \psi_i(\zeta_1, \tau_1) u(\zeta_1, t - \tau_1) d\zeta_1 d\tau_1$$

$$+ \int_0^\infty \int_0^\infty \sum_{i=1}^\infty \sum_{j=1}^\infty a_{ij}(x) \psi_i(\zeta_1, \tau_1) \psi_j(\zeta_2, \tau_2)$$

$$\times u(\zeta_1, t - \tau_1) u(\zeta_2, t - \tau_2) d\zeta_1 d\tau_1 d\zeta_2 d\tau_2 + \cdots$$

(39)

With the kernels given in terms of the orthonormal functions $\psi_i(\zeta, \tau)$, the following continuous-time form of spatio-temporal Volterra series will be obtained:

$$y(x, t) = \sum_{i=1}^\infty \int_0^\infty \cdots \int_0^\infty g_i(x, \zeta_1, \ldots, \zeta_r, \tau_1, \ldots, \tau_r)$$

$$\times \prod_{j=1}^r u(\zeta_j, t - \tau_j) d\zeta_1 d\tau_1 \cdots d\zeta_r d\tau_r.$$  

(40)

Appendix B. Karhunen–Loève expansion

Karhunen–Loève expansion (also known as principal component analysis) is to find an optimal basis from a representative set of process data. Suppose we have a set of observations $(y(x_t), t)\in \Omega, j = 1, \ldots, n_j, t = 1, \ldots, n_t$ (called snapshots). The problem is how to compute the most characteristic structure $\phi(x)$ among these snapshots $y(x, t)$.

B.1. Spatial correlation method

Actually this problem can be formulated as the one of obtaining a function $\phi(x)$ that maximizes the following objective function [11]:

$$\text{maximize} \quad \frac{\langle (\phi(x), y(x, t))^2 \rangle}{\langle \phi(x), \phi(x) \rangle}$$

subject to $\langle \phi(x), \phi(x) \rangle = 1$, $\phi(x) \in L^2(\Omega)$,

$$J = \langle (\phi(x), y(x, t))^2 \rangle - \lambda \langle \phi(x), \phi(x) \rangle,$$

(41)

and necessary condition for extrema is that the functional derivative vanishes for all variation $\phi + \delta \psi \in L^2(\Omega)$, where $\delta$ is a real number:

$$\frac{d\langle (\phi + \delta \psi), (\phi + \delta \psi) \rangle}{d\delta} (\delta = 0) = 0, \quad \langle \phi, \phi \rangle = 1.$$

Using the definitions of inner product and ensemble average, computing $d\langle (\phi + \delta \psi), (\phi + \delta \psi) \rangle/d\delta (\delta = 0)$, and using that $\phi(x)$ is an arbitrary function, the following necessary conditions for optimality can be obtained:

$$\int_\Omega R(x, \zeta) \phi_i(\zeta) d\zeta_i = \lambda_i \phi_i(x), \quad \langle \phi, \phi \rangle = 1.$$  

(42)

where $R(x, \zeta) = (y(x, t), y(\zeta, t))$ is spatial two-point correlation function, $\phi_i(x)$ is the $i$th eigenfunction, $\lambda_i$ is the corresponding eigenvalue.

Since the data are always discrete in space, one must solve numerically the integral Eq. (42). Discretizing the integral equation gives a $n_x \times n_x$ matrix eigenvalue problem. Thus, at most $n_x$ eigenfunctions at $n_t$ sampled spatial locations can be obtained. Then one can use the curve/surface fitting method [31,20] interpolate the eigenfunctions to locations where the data are not available.

B.2. Temporal correlation method

When $n_t$ is less than $n_x$, a computationally efficient way is provided by the method of snapshots [44,35] where the requisite eigenfunction $\phi(x)$ is expressed as a linear combination of the snapshots

$$\phi(x) = \sum_{i=1}^{n_x} \gamma_i y(x, t).$$

(43)

Substituting (43) into (42) gives the following eigenvalue problem:

$$\int_\Omega \frac{1}{n_t} \sum_{i=1}^{n_x} y(x, t) y(\zeta, t) \gamma_i d\zeta_i = \lambda_i \sum_{i=1}^{n_x} \gamma_i y(x, t).$$

(44)

Define a temporal two-point correlation matrix

$$Y_{r_t} = \frac{1}{n_t} \int_\Omega y(x, t) y(\zeta, t) d\zeta_i.$$  

Therefore, the $n_x \times n_x$ eigenvalue problem (44) can be reduced to a $n_t \times n_t$ problem as follows:

$$Y_{r_t} \gamma_i = \lambda_i \gamma_i.$$  

(45)

where $\gamma = [\gamma_1, \ldots, \gamma_{n_t}]$ is the $i$th eigenvector. The solution of the above eigenvalue problem yields the eigenvectors $\gamma_1, \ldots, \gamma_{n_t}$, which can be used in (43) to construct the eigenfunctions $\phi_1(x), \ldots, \phi_{n_x}(x)$. Because the matrix $Y$ is symmetric and positive semidefinite, thus
its eigenvalues $\lambda_i$ are real and non-negative. Furthermore, the computed eigenfunctions are orthogonal.

### 3. Dimension determination

The maximum number of nonzero eigenvalues is $n_k \leq \min(n_x, n_y)$. Let the eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_n$ and the corresponding eigenfunctions $\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x)$ be in the order of the magnitude of the eigenvalues. It can be proved that $[29]
\lambda_i = \langle (y(x,t), \varphi_i(x))^2 \rangle.

The eigenfunction that corresponds to the first eigenvalue is considered to be the most “energetic”. The total “energy” is defined as being the sum of the eigenvalues. To each eigenfunction, assign an “energy” percentage based on the associated eigenvalue

$$E_i = \frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i}.$$

Usually, the sufficient number of eigenfunctions that capture 99% of the system “energy” is used to determine the value of $n$. Experiences show that only the first few basis functions expansion can represent the dominant dynamics of many parabolic spatio-temporal systems. For an arbitrary set of basis functions \{\varphi_i(x)\}_{i=1}^{n}$, the following result holds $[29]
\sum_{i=1}^{n} \langle (y(x,t), \varphi_i(x))^2 \rangle = \sum_{i=1}^{n} \lambda_i \geq \sum_{i=1}^{n} \langle (y(x,t), \varphi_i(x))^2 \rangle.

It means that the Karhunen–Loève expansion is optimal on average in the class of representations by linear combination. That is why Karhunen–Loève expansion can give the lowest dimension $n$.

### C. Laguerre functions

Laguerre functions are defined as a functional series $[51]
\phi_i(t) \triangleq \sqrt{2p} \frac{e^{pt}}{(t-1)^{i/2} \Gamma(i/2)} e^{-2pt}, \quad i = 1, 2, \ldots, \infty, \quad p > 0,

where $p$ is the time-scaling factor, and $t \in [0, \infty)$ is a time variable. The Laplace transform of the $i$th Laguerre function is given by

$$\phi_i(s) = \sqrt{2p} \frac{(s - p)^{-i/2}}{(s + p)} e^{-2ps}, \quad i = 1, 2, \ldots, \infty, \quad p > 0.$$

Laguerre functions (46) and (47) form a complete orthonormal basis in the function space $L_2(0, \infty)$ and $H_2(C_1)$ respectively.

### D. State space realization

If $\Delta t$ is the sampling period and

$$\eta_1 = e^{-\beta_2 \Delta t}, \quad \eta_2 = \Delta t + \frac{2}{p}(e^{-\beta_2 \Delta t} - 1),$$

$$\eta_3 = -\Delta t e^{-\beta_2 \Delta t} - \frac{2}{p}(e^{-\beta_2 \Delta t} - 1), \quad \eta_4 = \sqrt{2p} \frac{(1 - e^{-\beta_2 \Delta t})}{p},

then

$$K_m = \begin{bmatrix}
\eta_1 & 0 & \cdots & 0 \\
\eta_2 & \eta_1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\eta_4 & \eta_3 & \cdots & \eta_1 \\
\end{bmatrix},$$

and

$$H_m = \begin{bmatrix}
\eta_4 & (-\eta_2/\Delta t)\eta_4 & \cdots & (-\eta_2/\Delta t)^{n_1-1}\eta_4 \\
\end{bmatrix}.$$

Using (16), the model output is given as follows:

$$y_n(x,t) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{q} \varphi_i(x) \hat{\theta}_{ijk}^y \sum_{w=1}^{m} \psi_{jw}(\xi) \hat{\psi}_{kwr}(t) + \cdots$$

Define $c_{kjw}(\xi) = \sum_{n=1}^{m} \sum_{l=1}^{n} \varphi_i(x) \delta_{ij} \hat{\psi}_{klw}(t)$ and $d_{kjw}(\xi) = \sum_{n=1}^{m} \sum_{l=1}^{n} \varphi_i(x) \delta_{ij} \hat{\psi}_{klw}(t)$, we have

$$y_n(x,t) = \sum_{w=1}^{m} \sum_{k=1}^{q} c_{kjw}(\xi) \hat{\psi}_{kwr}(t) + \cdots$$

Define

$$C_{kjw}(\xi) = \begin{bmatrix} c_{1kjw} & \cdots & c_{pqkw} \end{bmatrix}, \quad D_{kjw}(\xi) = \begin{bmatrix} d_{1kjw} & \cdots & d_{pqkw} \end{bmatrix},$$

then we have

$$y_n(x,t) = \sum_{w=1}^{m} \left[ C_{kjw}(\xi)^{\top} L_{kjw}(t) + \sum_{w=1}^{m} L_{kjw}(t)^{\top} D_{kjw}(\xi) L_{kjw}(t) \right] + \cdots$$

Eq. (48) can be further written in a simpler matrix form

$$y_n(x,t) = C(x)^{\top} L(t) + L(t)^{\top} D(x) L(t) + \cdots,$$

where

$$C(x) = \begin{bmatrix} C_1 \\ \vdots \\ C_m \end{bmatrix}, \quad D(x) = \begin{bmatrix} D_{11} & \cdots & D_{1m} \\ \vdots & \ddots & \vdots \\ D_{m1} & \cdots & D_{mm} \end{bmatrix}.$$
naturally leads to the convergence of the model to its optimum
\[ \hat{y}_n(x, t) \to y_0(x, t), \quad \text{w.p. 1 as } n_r \to \infty, \quad n_y \to \infty. \]

Define
\[ Q_{n_{n_{h}}}(\Theta) = \frac{1}{n_y} \sum_{j=1}^{n_y} \left\{ \sum_{t=1}^{n_t} e(x_j, t, \Theta)^2 \right\}. \]

As defined in (31), \( \Theta^* \) minimizes
\[ E(y(x, t) - V(x, t, \Theta, \{u(\zeta, \tau)\}))^2 = \lim_{n_t \to \infty} \left\{ \lim_{n_y \to \infty} E Q_{n_{n_{h}}}(\Theta) \right\}, \]

and the estimate \( \Theta^* \) minimizes \( Q_{n_{n_{h}}} \) as defined in (32).

In order to prove (52), we should prove the following convergence:
\[ \sup_{\Theta \in \mathcal{D}_0} |Q_{n_{n_{h}}}(\Theta) - E Q_{n_{n_{h}}}(\Theta)| \to 0, \quad \text{w.p. 1 as } n_t \to \infty, \quad n_y \to \infty. \]  

(53)

One feasible solution is to achieve the following convergence at any fixed spatial variable \( x \) before working at the spatio-temporal space.
\[ \sup_{\Theta \in \mathcal{D}_0} \sum_{t=1}^{n_t} |e(x, t, \Theta)^2 - E(e(x, t, \Theta)^2)| \to 0, \quad \text{w.p. 1 as } n_t \to \infty. \]

(54)

To achieve the convergence of (54), we have to obtain the convergence first at the pre-defined small open sphere, and then extend it to the global domain \( \mathcal{D}_0 \) using Heine–Bohr's theorem.

**E.1. Convergence of modeling error \( \epsilon \) to its optimum over \( B \)**

Define the supremum between the model error and its optimum as a random variable
\[ \eta(x, t) = [\eta(x, t, \Theta^0, \rho)] = \sup_{\Theta \in \mathcal{D}_0} [e(x, t, \Theta)^2 - E(e(x, t, \Theta)^2)]. \]

Let \( D \) be the open neighborhood of \( D_{\Theta} \) and choose \( \Theta^0 \in D_{\Theta} \). We can define a small open sphere centered at \( \Theta^0 \) as
\[ B(\Theta^0, \rho) = \{(\Theta) | \Theta - \Theta^0| \leq \rho \}. \]

Let \( B = B(\Theta^0, \rho) \cap D \), then
\[ \sup_{\Theta \in B} \sum_{t=1}^{n_t} |e(x, t, \Theta)^2 - E(e(x, t, \Theta)^2)| \leq \sum_{t=1}^{n_t} \eta(x, t). \]

(55)

Define \( \zeta(x, t) = \eta(x, t) - E \eta(x, t) \). If we can prove
- \( \zeta(x, t) \) satisfies Lemma 1 and
- the mean of \( \eta(x, t) \) is infinitesimal,

then \( \eta(x, t) \) is also infinitesimal.

Firstly, we consider
\[ |E(\zeta(x, t), \zeta(x, \tau))| = \text{Cov}[\eta(x, t), \eta(x, \tau)]. \]

Define \( \eta^0(x, t) = \sup_{\Theta \in B} [e^0(x, t, \Theta)^2 - E(e(x, t, \Theta)^2)] \), with
\[ e^0(x, t, \Theta) = y^0(x, t) - V(x, t, \Theta, \{u^0(\zeta, \tau)\}). \]

\[ \text{where } \{u^0(\zeta, \tau)\} \text{ denotes the input set } \{u^0(\zeta, \tau), \ldots, u^0(\zeta, \tau + 1), \ldots, 0\} \text{ for all } \zeta \in \Omega \text{.} \]

Hence
\[ \text{Cov}[\eta(x, t), \eta(x, \tau)] = \text{Cov}[\eta(x, t) - \eta^0(x, t), \eta(x, \tau)]. \]

Then using Schwartz's inequality, we have
\[ |E(\zeta(x, t), \zeta(x, \tau))| \leq |E[\eta(x, t)^2 \text{E}(\eta(x, t) - \eta^0(x, t))^2]|^{1/2}. \]

(56)

Since
\[ |\eta(x, t) - \eta^0(x, t)| \leq \sup_{\Theta \in B} |e(x, t, \Theta)^2 - e^0(x, t, \Theta)^2| \leq \sup_{\Theta \in B} \left\{ |e(x, t, \Theta)| + |e^0(x, t, \Theta)| \right\} \times \sup_{\Theta \in B} |e(x, t, \Theta)| - e^0(x, t, \Theta), \]

using Assumption 2, we can further have
\[ |\eta(x, t) - \eta^0(x, t)| \leq M \sum_{j=0}^{n_j} \left\{ |y(x, j)| + |y^0(x, j)| + \sup_{\zeta \in B} |u^0(\zeta, \tau)| + \sup_{\zeta \in B} |u(\zeta, \tau)| \right\} \]

\[ \times \sup_{\Theta \in B} |e(x, t, \Theta)| - e^0(x, t, \Theta)|. \]

(57)

Using Assumption 1 and Schwarz's inequality, we can finally derive
\[ E[\eta(x, t) - \eta^0(x, t)^2] \leq M \varepsilon^{-1}. \]

Following the similar derivation above and using Assumptions 2 and 1, we can also derive
\[ E\eta(x, t)^2 \leq M. \]

(58)

Placing (57) and (58) into (56), we can easily derive that \( \zeta(x, t) \) satisfies Lemma 1, that is
\[ \frac{1}{n_t} \sum_{t=1}^{n_t} \zeta(x, t) = \frac{1}{n_t} \sum_{t=1}^{n_t} (\eta(x, t) - E(\eta(x, t))) \to 0, \quad \text{w.p. 1 as } n_t \to \infty. \]

(59)

Secondly, the mean value of \( \eta \) can be expressed as,
\[ E[\eta(x, t)] = E \sup_{\Theta \in B} [e(x, t, \Theta)^2 - E(e(x, t, \Theta)^2)]. \]

Since the right-hand side is continuous with respect to \( \Theta \), \( E\eta(x, t) \) should be small if \( B \) is small. Furthermore, by Assumption 3,
\[ \left[ \frac{\partial e(x, t, \Theta)}{\partial \Theta} \right]^2 \leq M \left[ E(e(x, t, \Theta)^2) \right]^2 \leq M \sum_{j=0}^{n_j} \left\{ |y(x, j)| + \sup_{\zeta \in B} |u(\zeta, \tau)| \right\}^2, \]

where we again have used the uniform bounds in Assumption 2. Consequently, by Assumption 1,
\[ E \sup_{\Theta \in B} \left[ \frac{\partial e(x, t, \Theta)}{\partial \Theta} \right]^2 \leq M. \]

Now
\[ E\eta(x, t) = E \sup_{\Theta \in B} [e(x, t, \Theta)^2 - E(e(x, t, \Theta)^2)] \]
\[ \leq \sup_{\Theta \in B} \left[ E(e(x, t, \Theta)^2 - e(x, t, \Theta^0)^2) \right] + \sup_{\Theta \in B} \left[ \frac{\partial e(x, t, \Theta)^2}{\partial \Theta} \right] \]
\[ \leq \sup_{\Theta \in B} \left| E(e(x, t, \Theta)^2) \right| + \sup_{\Theta \in B} \left| \frac{\partial e(x, t, \Theta)^2}{\partial \Theta} \right| \]
\[ \times \sup_{\Theta \in B} |\Theta - \Theta^0| \leq M^2 \rho. \]

(60)
Finally from (60), (55) becomes
\[
\sup_{\omega \in D_0} \frac{1}{n_1} \sum_{t=1}^{n_1} |\epsilon(x, t, \Theta)|^2 \leq \frac{1}{n_1} \sum_{t=1}^{n_1} |\eta(x, t)| - E(x, t) \cdot M \rho. \tag{61}
\]
It is clearly seen from (59) that the first term of the right-hand side is arbitrarily small for a sufficiently large \( n_1 \). Since \( \rho \) can be arbitrarily small, therefore
\[
\sup_{\omega \in D_0} \frac{1}{n_1} \sum_{t=1}^{n_1} |\epsilon(x, t, \Theta)|^2 - E(x, t) \cdot M \rho \to 0, \text{ w.p. 1 as } n_1 \to \infty. \tag{62}
\]

### Appendix F. The 2nd-order spatio-temporal Volterra modeling

The 2nd-order spatio-temporal Volterra model can be expressed as follows:
\[
y(x, t) = \int_{\zeta_0}^{\zeta_1} g_1(x, t, \tau_1) u(t, \tau_1) d\tau_1 + \int_{\zeta_2}^{\zeta_3} g_2(x, t, \tau_2) u(t, \tau_2) d\tau_2,
\]
where \( g_1(\cdot) \) and \( g_2(\cdot) \) are the 1st-order and the 2nd-order spatio-temporal kernels. Then these kernels are time/space separated by expanding onto spatial output bases \( \{\phi_i(x)\}_{i=1}^m \), spatial input bases \( \{\psi_i(x)\}_{i=1}^m \), and temporal bases \( \{\phi_i(\tau)\}_{i=1}^m \):
\[
g_1(\cdot) = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \phi_i(x) \psi_j(x) \phi_{i+k}(\tau_1) \phi_{i+k}(\tau_1),
g_2(\cdot) = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \phi_i(x) \psi_j(x) \phi_{i+k}(\tau_1) \phi_{i+k}(\tau_2).
\]
After the parameters \( \theta_{ij}^{(1)} \) and \( \theta_{ij}^{(2)} \) are estimated using the algorithm in Section 3.3, the spatio-temporal kernels can be reconstructed using the time/space synthesis:
\[
g_1(\cdot) = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \phi_i(x) \psi_j(x) \phi_{i+k}(\tau_1),
g_2(\cdot) = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \phi_i(x) \psi_j(x) \phi_{i+k}(\tau_2).
\]


