A comprehensive expectation identification framework for multirate time-delayed systems

Jing Chen, Jie Gao, Yanjun Liu, Cheng Wang, Quanmin Zhu

***Abstract*—The expectation maximization (EM) algorithm has been extensively used to solve system identification problems with hidden variables. It needs to calculate a derivative equation and perform a matrix inversion in the EM-M step. The equations related to the EM algorithm may be unsolvable for some complex nonlinear systems, and the matrix inversion has heavy computational costs for large-scale systems. This paper provides two expectation based algorithms with the aim of constructing a comprehensive expectation framework concerning different kinds of time-delayed systems: (1) for a small-scale linear system, the classical EM algorithm can quickly obtain the parameter and time-delay estimates; (2) for a complex nonlinear system with low order, the proposed expectation gradient descent (EGD) algorithm can avoid derivative function calculation; (3) for a large-scale system, the proposed expectation multi-direction (EMD) algorithm does not require eigenvalue calculation and has less computational costs. These two algorithms are developed based on the gradient descent and multi-direction methods. Under such an expectation framework, different kinds of models are identified on a case-by-case basis. The convergence analysis and simulation examples show effectiveness of the algorithms.**

***Index Terms*—Time-delayed system, comprehensive expectation framework, maximization method, gradient descent method, multi-direction method**

I. Introduction

Parameter estimation is an important element in control engineering since it is the foundation for designing a controller particularly a robust controller [1], [2]. If a considered system does not have latent or unknown variables, e.g., unknown time-delay, missing outputs and unknown model identity, a number of well-known methods can be used for parameter estimation. These methods can be roughly divided into two classes: on-line algorithms and off-line algorithms [3], [4]. The on-line algorithms, such as the recursive least squares algorithm and the stochastic gradient algorithm, update the parameters in real time based on the latest collected data; thus, they are more sensitive to the new arrived data and may have slow convergence rates [5]. However, the off-line algorithms, e.g., the least squares algorithm and the gradient descent iterative algorithm, estimate the parameters through all the collected data. They are often more robust to the collected data and have fast convergence rates [6].

The last a few decades have witnessed the developments of communication networks and sensors. These developments have been utilized for many engineering applications. For example, in process control, sensors collect data and then transmit them to a control center through a network channel. When the network experiences congestion or a transmission problem, time-delay will occur [7], [8]. A special class of problems in time-delayed system identification is the fact that time-delay at each sampling instant is unknown and time varying. To estimate the parameters, the unknown time-varying time-delay at each sampling instant should be identified first, which makes identification of time-delayed systems significantly more demanding than standard system identification [9]. One of the most powerful off-line identification methods for systems with time-delay is the EM algorithm [10]. This algorithm is often used to identify systems with latent variables, such as missing outputs [11], unknown time-delays [12], hidden variables [13] and so on.

The basic procedure of the EM algorithm is to obtain the posterior distribution of the latent variables in the Expectation (E) step, and then to update the parameters using the maximization (MAX) method in M step [14], [15]. For example, Xie et al developed an EM algorithm for FIR models with varying time-delays, and the time-delays and parameters are iteratively estimated through the EM algorithm [12]. Zhao et al proposed a robust EM algorithm for ARX models with varying time-delays [16], [17]. Ma et al developed a modified Kalman filter based EM algorithm for ARX models with time-varying time-delays and missing outputs [18]. However, the MAX method should perform matrix inversion and solve a derivative equation, which brings two challenges: the computational costs increase significantly with increase of matrix dimension; the derived derivative equations often do not have analytical solutions [19]. Thus, the EM algorithm can be inefficient when the considered systems have one or more of the above problems.

Unlike the MAX method, the gradient descent (GD) method does not require matrix inversion and does not need to solve a derivative equation, and thus is commonly used for nonlinear system identification [20]. However, the GD method is relatively slow when the solution is close to the optimum. Technically, its asymptotic convergence rate is inferior to many other methods [21]. In addition, the GD algorithm requires computing the eigenvalue of a matrix in order to choose a suitable step-size [22]. A natural question arises: can we develop a method which can avoid calculating the eigenvalue and solving the derivative equation, but with less computational cost? To address this problem, the Arnoldis method which is often applied to solve the linear equation $Ax = b$ can be extended to solving the problems in EM and expectation GD (EGD) algorithms. Its basic idea is to generate several orthonormal directions in each iteration, by which a high-dimensional matrix can be transformed into a low-dimensional matrix [23]. In addition, it avoids solving a derivative equation.

To efficiently utilize the expectation based algorithms to solve complex identification problems, such as large-scale systems or nonlinear systems, there is a need to integrate new methods available in both machine learning and numerical mathematics to construct a more efficient expectation framework for handling different kinds of system identification problems. The focus of this paper is on applying the GD method and the multi-direction (MD) method to replace the MAX method in the EM algorithm to improve the estimation efficiency. The contributions are summarized as follows:

(1) The EGD algorithm does not require inverting a matrix and solving a derivative equation, thus can be applied to systems with more complex nonlinear characteristics.

(2) The expectation MD (EMD) algorithm avoids computing the

J. Chen and J. Gao are with School of Science, Jiangnan University, Wuxi 214122, PR China (chenjing1981929@126.com, gaojie@jiangnan.edu.cn).

Y.J. Liu and C. Wang are with Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, PR China (yanjunliu_1983@126.com, wangc@jiangnan.edu.cn).

Q.M. Zhu is with Department of Engineering Design and Mathematics, University of the West of England, Bristol BS16 1QY, UK (quan.zhu@uwe.ac.uk).

This work is supported by the National Natural Science Foundation of China (No. 61973137), and the Natural Science Foundation of Jiangsu Province (No. BK20201339)

eigenvalue and solving a derivative equation. Furthermore, it has less computational cost, therefore suitable for large-scale system identification.

(3) The EMD algorithm establishes a link between the EGD algorithm and the EM algorithm, thus constructs a more comprehensive expectation framework for different kinds of system identification problems.

The organization of this paper is as follows: Section II defines the time-delayed system of interest. Section III introduces the EM algorithm. Section IV proposes the EGD and EMD algorithms. The properties of all the considered algorithms are given in Section V. Three simulation examples are provided in Section VI. Section VII concludes this paper and points out possible future directions.

II. The time-varying time-delayed system

The time-delayed multirate (slow-rate output and fast-rate input) system to be considered in this paper is described as follows,

$$\begin{aligned} x(s) &= B(g)u(s), \\ y(S_i) &= x(S_i - \tau_i) + v(S_i), \end{aligned}$$

where s is the integer fast-rate sampling time index, S_i is the integer slow-rate sampling index (to be elaborately shortly), $u(s)$ is the measurable input and is taken as a persistent excitation signal, $x(s)$ is the true output, $y(S_i)$ is the output which is contaminated by the noise $v(S_i)$, $v(S_i)$ is a Gaussian white noise with variance σ^2, and τ_i is the unknown time-delay at the sampling instant S_i. The polynomial $B(g)$ is given as

$$B(g) := b_1 g^{-1} + b_2 g^{-2} + \cdots + b_m g^{-m},$$

in which $g^{-1}y(s) = y(s-1)$.

Some special considerations have to be made for the information vectors that will be defined shortly. We sample the output at the slow-rate sampling instant $S_i, i = 1, \cdots, N$, while the input data $u(s), s = 1, \cdots, L$ $(L > m)$ are sampled with smaller sampling period Δs. Thus, the slow-rate outputs are only measurable at $s = S_i \Delta s$. The time-delay $t_{d_i} = \tau_i \Delta s$ is time varying, and the integer τ_i is uniformly distributed between $[0, q]$, e.g., $p(\tau_i = j) = \frac{1}{q+1}, j = 0, \cdots, q$. It follows that the above time-delayed system can be rewritten as

$$y(S_i) = B(g)u(S_i - \tau_i) + v(S_i). \tag{1}$$

Define the information vector $\boldsymbol{\varphi}(S_i - \tau_i)$ and parameter vector $\boldsymbol{\theta}$ as

$$\begin{aligned} \boldsymbol{\varphi}(S_i - \tau_i) &= [u(S_i - \tau_i - 1), \cdots, u(S_i - \tau_i - m)]^{\mathrm{T}} \in \mathbb{R}^m, \\ \boldsymbol{\theta} &= [b_1, \cdots, b_m]^{\mathrm{T}} \in \mathbb{R}^m. \end{aligned}$$

Then, the following regression model is used to express the time-delayed system

$$y(S_i) = \boldsymbol{\varphi}^{\mathrm{T}}(S_i - \tau_i)\boldsymbol{\theta} + v(S_i). \tag{2}$$

Let $Y = \{y(S_1), y(S_2), \cdots, y(S_N)\}$, $U = \{u(1), u(2), \cdots, u(L)\}$ and the time-delays at each sampling instant S_i be $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_N\}$. Define the measurable data set $C_{obs} = \{Y, U\}$ and the hidden data set as $C_{mis} = \{\Gamma\}$. The purpose of the expectation based identification algorithms is to use the observed data C_{obs} to iteratively estimate the latent variables C_{mis} and the parameter vector $\boldsymbol{\theta}$.

III. The EM algorithm

The EM algorithm consists of E step and M step [12]. In the E-step, we compute the F-function based on the previous estimated parameter vector $\boldsymbol{\theta}_k$, where $\boldsymbol{\theta}_k$ is the parameter vector estimated in iteration k:

E-step

$$F(\boldsymbol{\theta}|\boldsymbol{\theta}_k) = \mathrm{E}_{C_{mis}|C_{obs},\boldsymbol{\theta}_k}\{\log p(C_{obs}, C_{mis}|\boldsymbol{\theta})\},$$

and then in the M-step, update the estimated parameter vector $\boldsymbol{\theta}$ by maximizing $F(\boldsymbol{\theta}|\boldsymbol{\theta}_k)$:

M-step

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} F(\boldsymbol{\theta}|\boldsymbol{\theta}_k).$$

Define

$$p(\tau_i = j|Y(S_i), U(S_i - \tau_i), \boldsymbol{\theta}_k) = w(\tau_i = j),$$

where $Y(S_i) = \{y(S_i), y(S_{i-1}), \cdots, y(S_1)\}$ and $U(S_i - \tau_i) = \{u(S_i - \tau_i - 1), u(S_i - \tau_i - 2), \cdots, u(1)\}$.

We can obtain the time-delay estimates in the E step, where the posterior distribution $p(\tau_i = j|y(S_i), \boldsymbol{\varphi}(S_i - \tau_i), \boldsymbol{\theta}_k)$ can be approximated by the following equation

$$\begin{aligned} &\hat{w}_{k+1}(\tau_i = j) = p(\tau_i = j|y(S_i), \boldsymbol{\varphi}(S_i - \tau_i), \boldsymbol{\theta}_k) \\ &= \frac{p(y(S_i)|\boldsymbol{\varphi}(S_i - \tau_i), \tau_i = j, \boldsymbol{\theta}_k) w(\tau_i = j)}{\sum_{j=0}^{q} p(y(S_i)|\boldsymbol{\varphi}(S_i - \tau_i), \tau_i = j, \boldsymbol{\theta}_k) w(\tau_i = j)}, \end{aligned} \tag{3}$$

in which $w(\tau_i = j)$ is the initial distribution of the time-delay τ_i and is equal to

$$w(\tau_i = j) = \frac{1}{q+1}.$$

It follows that Equation (3) can be simplified as

$$\hat{w}_{k+1}(\tau_i = j) = \frac{p(y(S_i)|\boldsymbol{\varphi}(S_i - \tau_i), \tau_i = j, \boldsymbol{\theta}_k)}{\sum_{j=0}^{q} p(y(S_i)|\boldsymbol{\varphi}(S_i - \tau_i), \tau_i = j, \boldsymbol{\theta}_k)}. \tag{4}$$

Once the time-delays are estimated, the F function can be written as

$$\begin{aligned} F(\boldsymbol{\theta}|\boldsymbol{\theta}_k) = \sum_{i=1}^{N}\sum_{j=0}^{q} \hat{w}_{k+1}(\tau_i = j) \log p(y(S_i)|\boldsymbol{\varphi}(S_i - \tau_i), \tau_i = j, \boldsymbol{\theta}) \times \\ p(u(S_i)|\tau_i = j, \boldsymbol{\theta}) p(\tau_i = j|\boldsymbol{\theta}), \end{aligned}$$

where $p(y(S_i)|\boldsymbol{\varphi}(S_i - \tau_i), \tau_i = j, \boldsymbol{\theta})$ is computed by

$$\begin{aligned} &p(y(S_i)|\boldsymbol{\varphi}(S_i - \tau_i), \tau_i = j, \boldsymbol{\theta}) = \\ &\frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-(y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta})^2}{2\sigma^2}\right], \end{aligned} \tag{5}$$

and $p(\tau_i = j|\boldsymbol{\theta}) = \frac{1}{q+1}$ and $p(u(S_i)|\tau_i = j, \boldsymbol{\theta})$ are both unrelated with the parameters. Therefore, we have

$$F(\boldsymbol{\theta}|\boldsymbol{\theta}_k) \propto -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=0}^{q} \hat{w}_{k+1}(\tau_i = j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}]^2. \tag{6}$$

The M step of the EM algorithm is to obtain an improved parameter vector estimate $\boldsymbol{\theta}_k$, in the sense:

$$F(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k) \geqslant F(\boldsymbol{\theta}_k|\boldsymbol{\theta}_k).$$

In the EM algorithm, maximizing the F function

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} F(\boldsymbol{\theta}|\boldsymbol{\theta}_k)$$

is equivalent to minimizing the following cost function

$$f(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=0}^{q} \hat{w}_{k+1}(\tau_i = j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}]^2, \tag{7}$$

and then the parameter estimates are computed by

$$\begin{aligned} \boldsymbol{\theta}_{k+1} = &\left[\sum_{i=1}^{N}\sum_{j=0}^{q} \hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\right]^{-1} \\ &\left[\sum_{i=1}^{N}\sum_{j=0}^{q} \hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)y(S_i)\right]. \end{aligned} \tag{8}$$

Equations (4) and (8) run iteratively until the parameter estimates converge.

Remark 1. In the EM algorithm, the parameter estimates are obtained using the maximization method, and in which $\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j) \in \mathbb{R}^{m\times m}$ [12], [15], [18]. When the order m is large, the computational costs of the matrix inversion in (8) are significantly large [24]. For example, when $m = 1000$, the flops of the matrix inversion are 10^9 (only counting the number of multiplication and division). Therefore, the EM algorithm is inefficient for large-scale system identification.

Remark 2. The majority of nonlinear equations are not solvable analytically, e.g., the generalized exponential autoregressive model in [4]. Thus, using the EM algorithm for developing nonlinear models whose parameters cannot be analytically extracted from the derivative function can be problematic; see Example 3 in Section VI.

IV. The EGD and EMD algorithms

The EM algorithm has two shortcomings: (1) when $\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j) \in \mathbb{R}^{m\times m}$, the flops of the matrix inversion are $O(m^3)$, that is, the computational costs increase sharply when m becomes larger; (2) the derivative equation of the cost function must have analytical solutions. To these ends, two approaches are employed in this section: one can avoid the derivative equation calculation and matrix inversion, and the other can transform a high-dimensional matrix inversion into a low-dimensional matrix inversion.

A. EGD algorithm

Rewrite the cost function

$$f(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}]^2.$$

Assume that the iterative parameter estimates are

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \lambda_k d_k, \tag{9}$$

where d_k is a direction and is computed by

$$-\frac{\nabla f(\boldsymbol{\theta})}{\nabla \boldsymbol{\theta}}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k} = \sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_k]. \tag{10}$$

Substituting Equation (10) into Equation (9) yields

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \lambda_k \sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j) \times [y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_k]. \tag{11}$$

Let the iteration function be

$$\boldsymbol{\theta}_{k+1} = F(\boldsymbol{\theta}_k) = [I - \lambda_k \sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)]\boldsymbol{\theta}_k + \lambda_k \sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)y(S_i). \tag{12}$$

For the sequence $\{\boldsymbol{\theta}_k\}$ to be convergent, one should ensure

$$\rho[I - \lambda_k \sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)] < 1,$$

where $\rho[G]$ means the spectral radius of matrix G [25]. Therefore, the step-size satisfies

$$0 < \lambda_k < \frac{2}{\lambda_{max}[\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)]}, \tag{13}$$

where $\lambda_{max}[\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)]$ means the largest eigenvalue of matrix $\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)$.

Remark 3. The EGD algorithm can ensure $f(\boldsymbol{\theta}_{k+1}) \leqslant f(\boldsymbol{\theta}_k)$. However, it cannot achieve the optimal parameter estimation in each iteration using current measured data and estimated latent variables (since it does not solve the derivative equation to get the optimal solution). Therefore, its convergence rate is slower than that of the EM algorithm.

Remark 4. The EGD algorithm can avoid calculating matrix inversion and solving a derivative equation (see Equation (10)), but needs to find the largest eigenvalue of a matrix (see Equation (13)). It brings another issue: computing the eigenvalues of a high-dimensional matrix is challenging or even infeasible.

Remark 5. The EGD algorithm is a first-order iterative optimization algorithm for finding the minimum of the cost function $f(\boldsymbol{\theta})$, which means that it is easy to fall into a local optimum.

B. EMD algorithm

The key to the EGD algorithm is to update the parameters using only one direction in each iteration, which performs the matrix eigenvalue calculation instead of the matrix inversion but with cost of slow convergence rates. In this subsection, an EMD algorithm is developed, which updates the parameters based on l $(l < m)$ directions in each iteration. Therefore, it has faster convergence rates than the EGD algorithm, and has less computational loads than the EM algorithm.

Arnoldi's method [23]: Assuming that the Krylov subspace is $\bar{\boldsymbol{\xi}}(l) = \mathrm{span}\{\bar{\xi}(1), Q\bar{\xi}(1), \cdots, Q^{l-1}\bar{\xi}(1)\}$, where $Q \in \mathbb{R}^{m\times m}$ is nonsingular, $\bar{\xi}(1) \in \mathbb{R}^m$ and satisfies $\|\bar{\xi}(1)\| = 1$, then, an orthonormal basis $\{\bar{\xi}(1), \bar{\xi}(2), \cdots, \bar{\xi}(l)\}$ can be computed by the following steps.

1) INITIALIZATION. Given a vector $\bar{\xi}(1)$ with $\|\bar{\xi}(1)\| = 1$.
2) Let $i = 1$.
3) Compute $(Q\bar{\xi}(i), \bar{\xi}(j))$, $j = 1, 2, \cdots, i$.
4) Compute $\xi(i+1) = Q\bar{\xi}(i) - \sum_{j=1}^{i}(Q\bar{\xi}(i), \bar{\xi}(j))\bar{\xi}(j)$.
5) Normalize $\bar{\xi}(i+1) = \frac{\xi(i+1)}{\|\xi(i+1)\|}$.
6) Let $i = i + 1$ and go to step 3),

where (a, b) means the inner product of vectors a and b.

Assuming that the parameter vector estimate in iteration k is $\boldsymbol{\theta}_k$ and the time-delay estimates are $\{\tau_1^{k+1}, \tau_2^{k+1}, \cdots, \tau_N^{k+1}\}$ (which have been estimated in the E step based on the parameter vector estimate $\boldsymbol{\theta}_k$).

The residual error can be written by

$$\xi^{k+1}(1) = \sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_k]. \tag{14}$$

Let

$$\bar{\xi}^{k+1}(1) = \frac{\xi^{k+1}(1)}{\|\xi^{k+1}(1)\|} = \frac{\xi^{k+1}(1)}{\beta^{k+1}},$$
$$Q_{k+1} = \sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j).$$

Based on Arnoldi's method, we construct the following l orthonormal directions as

$$\bar{\boldsymbol{\xi}}^{k+1}(l) = [\bar{\xi}^{k+1}(1), \bar{\xi}^{k+1}(2), \cdots, \bar{\xi}^{k+1}(l)], \tag{15}$$

where $l < m$, and $[\bar{\xi}^{k+1}(i)]^{\mathrm{T}}\bar{\xi}^{k+1}(j) = 0, i \neq j$.

It follows that the parameter estimates in iteration $k+1$ are computed by

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \bar{\boldsymbol{\xi}}^{k+1}(l)\gamma_k, \tag{16}$$

where $\bar{\boldsymbol{\xi}}^{k+1}(l)$ contains l orthonormal directions/vectors. Next, we aim to find a suitable γ_k which can ensure

$$\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}] \leqslant$$
$$\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k}].$$

Substituting Equation (16) into the left side of the above equation yields

$$\begin{aligned}&\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}]\\ &= \beta^{k+1}\bar{\xi}^{k+1}(1) -\\ &\quad [Q_{k+1}\bar{\xi}^{k+1}(1), Q_{k+1}\bar{\xi}^{k+1}(2), \cdots, Q_{k+1}\bar{\xi}^{k+1}(l)]\gamma_k. \end{aligned}\tag{17}$$

According to Arnoldi's method, we have

$$Q_{k+1}\bar{\boldsymbol{\xi}}^{k+1}(l) = \bar{\boldsymbol{\xi}}^{k+1}(l+1)P_k, \tag{18}$$

$$P_k = \begin{bmatrix} \bar{P}_k \\ p^k_{l+1,l}e^{\mathrm{T}}(l) \end{bmatrix} \in \mathbb{R}^{(l+1)\times l}, \tag{19}$$

$$\bar{P} = \begin{bmatrix} p^k_{1,1} & p^k_{1,2} & \cdots & p^k_{1,l-1} & p^k_{1,l} \\ p^k_{2,1} & p^k_{2,2} & \cdots & p^k_{2,l-1} & p^k_{2,l} \\ 0 & p^k_{3,2} & \cdots & p^k_{3,l-1} & p^k_{3,l} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & p^k_{l,l-1} & p^k_{l,l} \end{bmatrix} \in \mathbb{R}^{l\times l}, \tag{20}$$

$$e_l(l) = [0, 0, \cdots, 0, 1]^{\mathrm{T}} \in \mathbb{R}^l, \tag{21}$$

where $p^k_{j,i} = (Q_{k+1}\bar{\xi}^{k+1}(i), \bar{\xi}^{k+1}(j)), j \leqslant i$ and $p^k_{i+1,i} = \|\xi^{k+1}_{i+1}\|$, the subscript l in $e_l(l)$ means that the order of the vector is l, and l in the brackets means that the lth element in the vector is 1.

Equation (17) is transformed into

$$\begin{aligned}&\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}]\\ &= \beta^{k+1}\bar{\xi}^{k+1}(1) - \bar{\boldsymbol{\xi}}^{k+1}(l+1)P_k\gamma_k\\ &= \bar{\boldsymbol{\xi}}^{k+1}(l+1)[\beta^{k+1}e_{l+1}(1) - P_k\gamma_k].\end{aligned}$$

Since $\bar{\boldsymbol{\xi}}^{k+1}(l+1) \in \mathbb{R}^{m\times(l+1)}$ and $m > (l+1)$, the following equality holds

$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}]\| =$$
$$\|\beta^{k+1}e_{l+1}(1) - P_k\gamma_k\|. \tag{22}$$

According to matrix theory, there exists an orthogonal matrix $J \in \mathbb{R}^{(l+1)\times(l+1)}$, which can make Equation (22) satisfy

$$\begin{aligned}&\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}]\| =\\ &\|J^{\mathrm{T}}\|\|J\beta^{k+1}e_{l+1}(1) - JP_k\gamma_k\| =\\ &\left\|\left[\begin{bmatrix}\boldsymbol{\eta}^{\mathrm{T}} \\ \eta_{k+1}\end{bmatrix} - \begin{bmatrix}\tilde{P}_k \\ \mathbf{0}\end{bmatrix}\gamma_k\right]\right\|,\end{aligned}\tag{23}$$

where

$$J\beta^{k+1}e_{l+1}(1) = \begin{bmatrix}\boldsymbol{\eta}^{\mathrm{T}} \\ \eta_{k+1}\end{bmatrix},$$

$$JP_k = \begin{bmatrix}\tilde{P}_k \\ \mathbf{0}\end{bmatrix},$$
$$\boldsymbol{\eta}^{\mathrm{T}} \in \mathbb{R}^l, \quad \tilde{P}_k \in \mathbb{R}^{l\times l}, \quad \mathbf{0} \in \mathbb{R}^{1\times l}.$$

Then, the following theorem is obtained.

Theorem 1: Assume that the parameter vector estimate $\boldsymbol{\theta}_{k+1}$ using the EMD algorithm is expressed by (16), the l directions $\bar{\boldsymbol{\xi}}^{k+1}(l)$ are written by (15). When $\gamma_k = [\tilde{P}_k]^{-1}\boldsymbol{\eta}^{\mathrm{T}}$, the following inequality holds

$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}]\| \leqslant$$
$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k}]\|.$$

(Proof is given in Appendix A.)

Remark 6. Theorem 1 shows that the EMD method can keep the cost function $f(\boldsymbol{\theta})$ monotonically decreasing, and a larger l leads to a smaller cost function. When $l < m$, the EMD algorithm performs an $l - order$ matrix inversion calculation instead of an $m - order$ inversion calculation. Thus, its computational efforts are less than those of the EM algorithm.

Remark 7. The EMD algorithm updates the parameters using l directions rather than 1 direction in each iteration. Thus, its convergence rates are faster than those of the EGD algorithm. In addition, it does not require the eigenvalue calculating.

C. Comprehensive expectation framework

When the parameter estimates are obtained by using the above two methods, the time-delay and the parameter vector estimates will be iteratively updated until they both converge.

The comprehensive expectation framework then is summarized as follows.

1) Update the unknown variables

Expectation step:

$$\hat{w}_{k+1}(\tau_i = j) = \frac{p(y(S_i)|\boldsymbol{\varphi}(S - \tau_i), \tau_i = j, \boldsymbol{\theta}_k)}{\sum_{j=0}^{q} p(y(S_i)|\boldsymbol{\varphi}(S_i - \tau_i), \tau_i = j, \boldsymbol{\theta}_k)}. \tag{24}$$

2) Estimate the parameters

2-A: Maximization method

$$\begin{aligned}\boldsymbol{\theta}_{k+1} = &[\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)]^{-1}\\ &[\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)y(S_i)].\end{aligned}\tag{25}$$

2-B: Gradient descent method

$$\begin{aligned}\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \lambda_k\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j) \times\\ [y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_k].\end{aligned}\tag{26}$$

2-C: Multi-direction method

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \bar{\boldsymbol{\xi}}^{k+1}(l)[\tilde{P}_k]^{-1}\boldsymbol{\eta}^{\mathrm{T}}. \tag{27}$$

Remark 8. The differences among these three algorithms are resulted from different methods for estimating the parameters. The EM algorithm performs MAX method, the EGD algorithm utilizes GD method, and the EMD algorithm uses MD method.

V. Properties of the EGD and EMD algorithms

In this section, some properties of the EGD and EMD algorithms are given.

A. Convergence analysis

For a random sampling instant S_i, we introduce a proposal distribution $q(\tau_i)$ and define

$$F(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(y(S_i)|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \frac{p(y(S_i), \tau_i|\boldsymbol{\theta})}{p(\tau_i|y(S_i), \boldsymbol{\theta})} = F_1(\boldsymbol{\theta}) - F_2(\boldsymbol{\theta}) + F_3(\boldsymbol{\theta}), \quad (28)$$

where

$$F_1(\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{j=0}^{q} q(\tau_i = j) \log p(y(S_i), \tau_i = j|\boldsymbol{\theta}),$$

$$F_2(\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{j=0}^{q} q(\tau_i = j) \log q(\tau_i = j),$$

$$F_3(\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{j=0}^{q} q(\tau_i = j) \log \frac{q(\tau_i = j)}{p(\tau_i = j|y(S_i), \boldsymbol{\theta})}.$$

The following theorem can be obtained.

Theorem 2: Assume that the time-delay and parameter vector estimates using the EGD algorithm are updated by Equations (24) and (26), while the estimates using the EMD algorithm are determined by Equations (24) and (27). Then, the function $F(\boldsymbol{\theta})$ expressed by (28) is monotonically increasing.
(The detailed proof is given in Appendix B.)

Remark 9. Since the function $F(\boldsymbol{\theta})$ is monotonically increasing, both the EGD and EMD algorithms are convergent.

B. The relationships of the three algorithms

Case 1: When the order of the direction is $l = m$, namely

$$\bar{\boldsymbol{\xi}}^{k+1}(m) = [\bar{\xi}^{k+1}(1), \bar{\xi}^{k+1}(2), \cdots, \bar{\xi}^{k+1}(m)],$$

then, $\bar{\boldsymbol{\xi}}^{k+1}(m)$ is equivalent to $\bar{\boldsymbol{\xi}}^{k+1}(m+1)$.

According to Equation (17), the step-size $[\tilde{P}_k]^{-1}\boldsymbol{\eta}^{\mathrm{T}}$ of the EMD algorithm can be computed by

$$[\tilde{P}_k]^{-1}\boldsymbol{\eta}^{\mathrm{T}} = [Q_{k+1}\bar{\boldsymbol{\xi}}^{k+1}(m)]^{-1}\beta^{k+1}\bar{\xi}^{k+1}(1).$$

Substituting the above equation into Equation (27) yields

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \bar{\boldsymbol{\xi}}^{k+1}(m)[Q_{k+1}\bar{\boldsymbol{\xi}}^{k+1}(m)]^{-1}\beta^{k+1}\bar{\xi}^{k+1}(1) = [\sum_{i=1}^{N}\sum_{j=0}^{q} \hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)\boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)]^{-1} [\sum_{i=1}^{N}\sum_{j=0}^{q} \hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)y(S_i)],$$

which is the same as Equation (25). In this case, the EMD algorithm is equivalent to the EM algorithm. There is no need to use the EMD algorithm to replace the EM algorithm, because the direction calculation in each iteration of the EMD algorithm would lead to heavier computational efforts than those of the EM algorithm.

Case 2: When the order of the direction is $l = 1$, according to Equations (31) and (32) in Appendix, we get

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \frac{p_{1,1}^k}{\sqrt{(p_{1,1}^k)^2 + (p_{2,1}^k)^2}} \sum_{i=1}^{N}\sum_{j=0}^{q} \hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j) \times [y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)].$$

If the step-size of the EGD algorithm is $\lambda_k = \frac{p_{1,1}^k}{\sqrt{(p_{1,1}^k)^2 + (p_{2,1}^k)^2}}$, the EMD algorithm can be regarded as an EGD algorithm.

Remark 10. The EMD algorithm can be regarded as an EGD algorithm when $l = 1$, but it does not require the eigenvalue calculation in each iteration. For this reason, the EMD algorithm can be applied to large-scale system identification.

Remark 11. The EMD algorithm acts as a bridge between the EM algorithm and the EGD algorithm: (1) if $l = 1$, the EMD algorithm is equivalent to the EGD algorithm; (2) if $l = m$, the EMD algorithm is the same as the EM algorithm.

VI. Examples

A. Example 1

Consider the following time-delayed system,

$$x(s) = b_1u(s-1) + b_2u(s-2) + b_3u(s-3) + b_4u(s-4) + b_5u(s-5) + b_6u(s-6),$$
$$y(S_i) = x(S_i - \tau_i) + v(S_i),$$
$$\boldsymbol{\theta} = [b_1, b_2, b_3, b_4, b_5, b_6]^{\mathrm{T}} = [0.2, 0.4, -0.3, 0.2, 0.7, 0.6]^{\mathrm{T}},$$
$$u \sim N(0,1), \quad v \sim N(0, 0.01), \quad \tau_i \in \{0, 1, 2, 3\}.$$

In the simulation, we sample $L = 800$ data points for fast-rate inputs, and sample slow-rate outputs at every four fast-rate sampling intervals, e.g., $y(S_1 = 4)$, $y(S_2 = 8), \cdots, y(S_{200} = 800)$ are sampled, while the other outputs are missing, and the initial probabilities $p(\tau_i = j) = 0.25, i = 1, 2, \cdots, 200, j = 0, 1, 2, 3$.

First, apply the EM, EGD and EMD ($l = 4$) algorithms to estimate the parameters of the time-delayed system, the parameter estimates and their estimation errors $\delta := \|\boldsymbol{\theta}_k - \boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ versus k are shown in Fig. 1. The rates of correct identification of time-delays are shown in Table I.

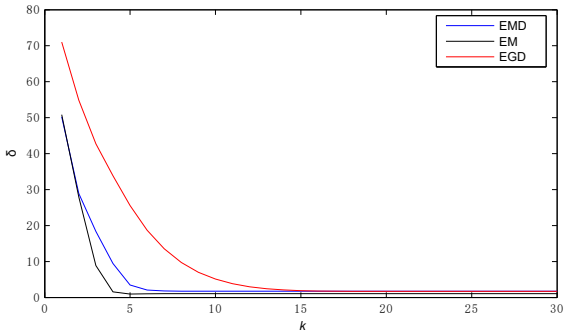


Fig. 1. The parameter estimation errors δ versus k

TABLE I
THE CORRECT IDENTIFICATION RATES OF THE UNKNOWN TIME-DELAYS

Algorithm	EM	EGD	EMD
Rate	88.5%	85%	87%

From this simulation example, we have the following observations:
(1) The parameter and time-delay estimates asymptotically converge to their true values by using these three algorithms, which can be shown in Fig. 1 and Table I.
(2) Fig. 1 shows that the EM algorithm has the fastest convergence rate, the second is the EMD algorithm, and the slowest is the EGD algorithm.

Second, we use the EMD algorithm with different directions ($l = 1, 2, 3, 5$) to estimate the parameters of the time-delayed system. The estimation errors $\delta := \|\boldsymbol{\theta}_k - \boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ versus k are shown in Fig. 2. From Fig. 2, we can see that a larger number of directions lead to a faster convergence rate.

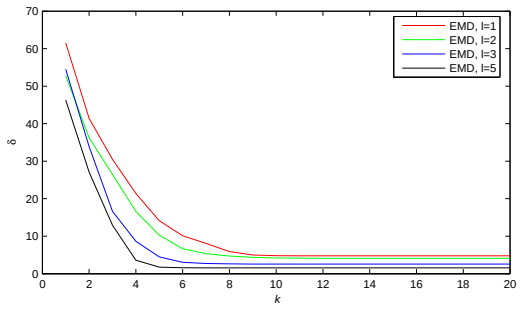


Fig. 2. The parameter estimation errors δ versus k

B. Example 2

Consider the following $15th$-order time-delayed system,

$$
\begin{aligned}
&x(s) = b_1u(s-1) + b_2u(s-2) + \cdots + b_{15}u(s-15),\\
&y(S_i) = x(S_i - \tau_i) + v(S_i),\\
&\boldsymbol{\theta} = [b_1, b_2, \cdots, b_{15}]^{\mathrm{T}} = [0.2, 0.4, -0.3, 0.2, 0.7, 0.6, -0.3,\\
&\qquad 0.2, 0.3, 0.1, 0.3, 0.2, -0.4, 0.5, -0.6]^{\mathrm{T}},\\
&u \sim N(0,1), \quad v \sim N(0, 0.1^2), \quad \tau_i \in \{0, 1, 2\}.
\end{aligned}
$$

We sample $L = 900$ data points for fast-rate inputs, and sample slow-rate outputs at every three fast-rate sampling intervals, that is, $L/3 = 3$. Assign the initial probabilities as $p(\tau_i = j) = \frac{1}{3}, j = 0, 1, 2, i = 1, 2, \cdots, 300$.

Apply the EM, EGD and EMD ($l = 4$) algorithms to the time-delayed systems. The parameter estimation errors are shown in Fig. 3, and the time-delay estimates are shown in Fig. 4. In Fig. 4, the true time-delays are described by the blue asterisks, while the time-delay estimates are expressed by the red circles. It shows that the time-delay estimates by using these three algorithms can catch the true values.

Finally, the Monte Carlo simulations (with 300 different noise seeds) are also performed. The elapsed time of these three algorithms by using the Monte Carlo method is displayed in Table II (**by Intel(R) Core(TM) i5-7220U: 2.50GHz, 2.71GHz; RAM: 8.0 GB; Windows 10**), which indicates that the EMD algorithm has the shortest elapsed time.

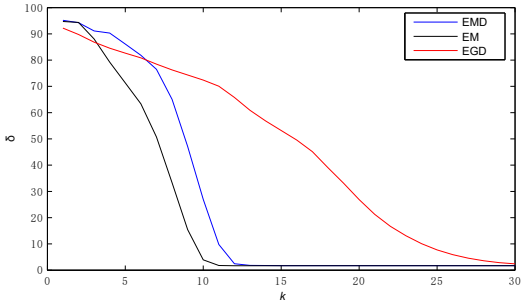


Fig. 3. The parameter estimation errors δ versus k

TABLE II
THE ELAPSED TIMES

Algorithm	EM	EGD	EMD ($l = 4$)
Time (second)	138.32	156.74	102.53

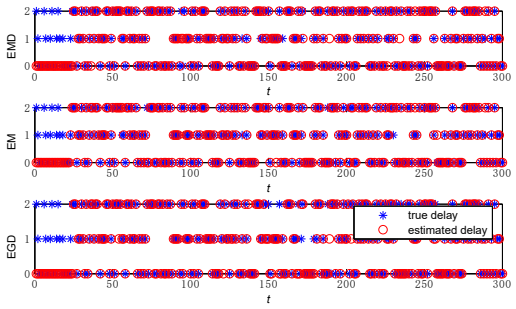


Fig. 4. The time-delay estimates

C. Example 3

Consider an exponential nonlinear (EN) model as follows,

$$
\begin{aligned}
&x(s) = e^{b_1u(s-1)}u(s-1) + b_2u(s-2),\\
&y(S_i) = x(S_i - \tau_i) + v(S_i),\\
&\boldsymbol{\theta} = [b_1, b_2]^{\mathrm{T}} = [0.2, 0.4]^{\mathrm{T}},\\
&u \sim N(0,1), \quad v \sim N(0, 0.1^2), \quad \tau_i \in \{0, 1, 2\}.
\end{aligned}
$$

Define the cost function as

$$
\begin{aligned}
f(\boldsymbol{\theta}) = f(b_1, b_2) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j) \times\\
[y(S_i) - e^{b_1u(S_i-j-1)}u(S_i - j - 1) - b_2u(S_i - j - 2)]^2.
\end{aligned}
$$

Using the EM algorithm yields

$$
\begin{bmatrix}
\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)e^{b_1u(S_i-j-1)}e(S_i)\\
\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)u(S_i - j - 2)e(S_i)
\end{bmatrix} = \begin{bmatrix} 0\\ 0 \end{bmatrix},
$$

where $e(S_i) = [y(S_i) - e^{b_1u(S_i-j-1)}u(S_i - j - 1) - b_2u(S_i - j - 2)]$. Clearly, the above equation does not have an analytical solution, which means that the EM algorithm is inefficient for this EN model.

Assume that the parameter estimates in iteration $k-1$ are $\boldsymbol{\theta}_{k-1} = [b_1^{k-1}, b_2^{k-1}]^{\mathrm{T}}$, the residual error in iteration k is written by

$$
\xi^k(1) = \begin{bmatrix}
\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)e^{b_1^{k-1}u(S_i-j-1)}e^{k-1}(S_i)\\
\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)u(S_i - j - 2)e^{k-1}(S_i)
\end{bmatrix},
$$

where $e^{k-1}(S_i) = [y(S_i) - e^{b_1^{k-1}u(S_i-j-1)}u(S_i - j - 1) - b_2^{k-1}u(S_i - j - 2)]$. Then, the EGD and EMD algorithms can be applied to this EN model.

We sample $L = 300$ data points for fast-rate inputs, and sample slow-rate outputs at every three fast-rate sampling intervals. Assign the initial probabilities as $p(\tau_i = j) = \frac{1}{3}, j = 0, 1, 2, i = 1, 2, \cdots, 100$. Use the EMD algorithm for the time-delayed EN model ($l = 1$). The parameter estimates and their estimation errors are shown in Fig. 5. The time-delay estimates are shown in Fig. 6. Figs. 5 and 6 show that the EMD algorithm is efficient for nonlinear systems with complex structures.

The Monte Carlo simulations (with 100 different noise seeds) are also performed based on the EGD and EMD algorithms. The boxplot of parameter estimates of different noise seeds are shown in Fig. 7. Fig. 7 shows that the EMD algorithm has more accurate parameter estimates (the red lines in the blue box of the EMD algorithm are much closer to the true values than those of the EGD algorithm) and smaller estimation variances (the EMD algorithm has less estimates beyond the blue box than those of the EGD algorithm). The elapsed times of these two kinds of algorithms are shown in Table III.

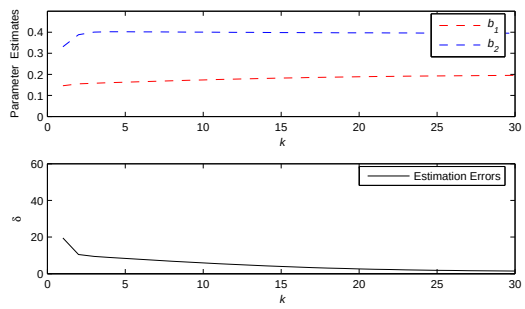


Fig. 5. The parameter estimates and their estimation errors

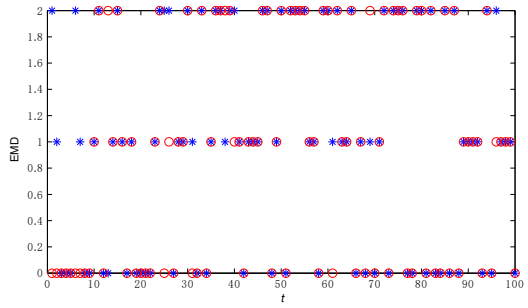


Fig. 6. The time-delay estimates

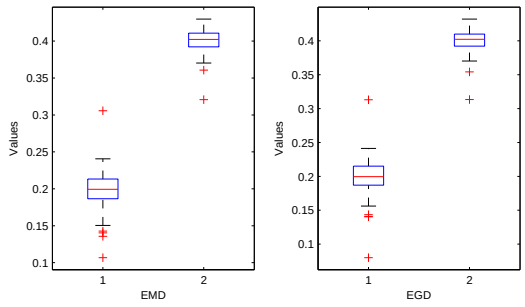


Fig. 7. The parameter estimates using EMD and EGD algorithms for 100 different noise seeds

TABLE III
THE ELAPSED TIMES

Algorithm	EMD ($l = 1$)	EGD
Time (second)	18.576	27.030

This example shows that both the EMD and EGD algorithms can identify systems with complex nonlinear structures, but the EM algorithm cannot. In addition, the EMD algorithm is robust to noise, and the EMD algorithm has less elapsed times than those of the EGD algorithm.

VII. Conclusions

Two expectation based identification algorithms are proposed for time-varying time-delayed systems in this paper. The EGD algorithm obtains the parameter estimates using the gradient descent method, which can avoid the matrix inversion and solving derivative equations. Therefore, it can be applied to solving complex nonlinear system identification problems. The EMD algorithm updates the parameter estimates through the multi-direction method. It does not require matrix eigenvalue calculation and can transform a high-dimensional matrix inversion into a lower-dimensional matrix inversion; thus, it can be applied to solving large-scale system identification problems. Convergence analysis and simulation examples demonstrate efficiency of these two algorithms. These two algorithms combining the EM algorithm construct a comprehensive expectation identification framework for systems with hidden variables, and we can choose the optimal algorithm for a considered model on a case by case basis.

In addition, the EMD algorithm builds a link between the EM algorithm and EGD algorithm: when $l = 1$, the EMD algorithm can be regraded as an EGD algorithm; when $l = m$, the EMD algorithm is equivalent to the EM algorithm. Therefore, these three algorithms should be applicable to a broad class of systems, including systems with other kind of latent variables (such as hidden state variables), large-scale systems and nonlinear systems, and thus they offer various options for different systems.

Appendix A

Proof of Theorem 1. Assuming that $l = 1$, we have

$$\bar{\boldsymbol{\xi}}^{k+1}(1) = [\bar{\xi}^{k+1}(1)],$$

$$\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_k] = \xi^{k+1}(1),$$

$$Q_{k+1}\bar{\boldsymbol{\xi}}^{k+1}(1) = [\bar{\xi}^{k+1}(1), \bar{\xi}^{k+1}(2)]\begin{bmatrix} p_{1,1}^k \\ p_{2,1}^k \end{bmatrix}.$$

The orthogonal matrix $J \in \mathbb{R}^{2\times 2}$ is written by

$$J = \begin{bmatrix} \frac{p_{1,1}^k}{\sqrt{(p_{1,1}^k)^2+(p_{2,1}^k)^2}} & \frac{p_{2,1}^k}{\sqrt{(p_{1,1}^k)^2+(p_{2,1}^k)^2}} \\ \frac{-p_{2,1}^k}{\sqrt{(p_{1,1}^k)^2+(p_{2,1}^k)^2}} & \frac{p_{1,1}^k}{\sqrt{(p_{1,1}^k)^2+(p_{2,1}^k)^2}} \end{bmatrix}. \tag{29}$$

It follows that

$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}^1]\| =$$

$$\left\| J\begin{bmatrix} \beta^{k+1} \\ 0 \end{bmatrix} - J\begin{bmatrix} p_{1,1}^k \\ p_{2,1}^k \end{bmatrix}\gamma_k \right\|, \tag{30}$$

where

$$\boldsymbol{\theta}_{k+1}^1 = \boldsymbol{\theta}_k + \bar{\boldsymbol{\xi}}^{k+1}(1)\gamma_k = \boldsymbol{\theta}_k + \bar{\xi}^{k+1}(1)\gamma_k, \tag{31}$$

and the superscript 1 in the parameter vector estimate $\boldsymbol{\theta}_{k+1}^1$ means that the number of direction is 1. Substituting Equation (29) into Equation (30) yields

$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}^1]\| =$$

$$\left\| \begin{bmatrix} \frac{p_{1,1}^k\beta^{k+1}}{\sqrt{(p_{1,1}^k)^2+(p_{2,1}^k)^2}} \\ \frac{-p_{2,1}^k\beta^{k+1}}{\sqrt{(p_{1,1}^k)^2+(p_{2,1}^k)^2}} \end{bmatrix} - \begin{bmatrix} \sqrt{(p_{1,1}^k)^2+(p_{2,1}^k)^2}\gamma_k \\ 0 \end{bmatrix} \right\|.$$

Then, the step-size can be computed by

$$\gamma_k = \frac{p_{1,1}^k\beta^{k+1}}{(p_{1,1}^k)^2+(p_{2,1}^k)^2}, \tag{32}$$

which means

$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i = j)\boldsymbol{\varphi}(S_i - j)[y(S_i) - \boldsymbol{\varphi}^{\mathrm{T}}(S_i - j)\boldsymbol{\theta}_{k+1}^1]\| =$$

$$\left\| \frac{-p_{2,1}^k\beta^{k+1}}{\sqrt{(p_{1,1}^k)^2+(p_{2,1}^k)^2}} \right\|.$$

Then, we have

$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i=j)\boldsymbol{\varphi}(S_i-j)[y(S_i)-\boldsymbol{\varphi}^{\mathrm{T}}(S_i-j)\boldsymbol{\theta}_{k+1}^{1}]\| \leqslant$$
$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i=j)\boldsymbol{\varphi}(S_i-j)[y(S_i)-\boldsymbol{\varphi}^{\mathrm{T}}(S_i-j)\boldsymbol{\theta}_{k}]\|.$$

Following the same way, the following inequality holds

$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i=j)\boldsymbol{\varphi}(S_i-j)[y(S_i)-\boldsymbol{\varphi}^{\mathrm{T}}(S_i-j)\boldsymbol{\theta}_{k+1}]\| \leqslant$$
$$\|\sum_{i=1}^{N}\sum_{j=0}^{q}\hat{w}_{k+1}(\tau_i=j)\boldsymbol{\varphi}(S_i-j)[y(S_i)-\boldsymbol{\varphi}^{\mathrm{T}}(S_i-j)\boldsymbol{\theta}_{k}]\|,$$

in which $\boldsymbol{\theta}_{k+1}$ is estimated using l directions.

Appendix B

Proof of Theorem 2. According to Equation (4), the posterior distribution of τ_i in iteration k can be transformed into

$$q(\tau_i=j)=p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k). \tag{33}$$

Then, one can get

$$F(\boldsymbol{\theta}|\boldsymbol{\theta}_k)=F_1(\boldsymbol{\theta}|\boldsymbol{\theta}_k)-F_2(\boldsymbol{\theta}|\boldsymbol{\theta}_k)+F_3(\boldsymbol{\theta}|\boldsymbol{\theta}_k). \tag{34}$$

Clearly, the first function

$$F_1(\boldsymbol{\theta}|\boldsymbol{\theta}_k)=\sum_{i=1}^{N}\sum_{j=0}^{q}p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k)\times \log p(y(S_i),\tau_i=j|\boldsymbol{\theta})$$

is equivalent to the cost function in Equation (7). Based on Remark 4 and Theorem 1, both the EGD and EMD algorithms can guarantee that

$$F_1(\boldsymbol{\theta}_k|\boldsymbol{\theta}_k)\leqslant F_1(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k), \tag{35}$$

when the parameter estimates are computed by Equations (26) and (27), respectively.

On the other hand, the second function of Equation (34) is

$$F_2(\boldsymbol{\theta}|\boldsymbol{\theta}_k)=\sum_{i=1}^{N}\sum_{j=0}^{q}p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k)\times \log p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k),$$

which is a constant. Thus, it leads to

$$F_2(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k)=F_2(\boldsymbol{\theta}_k|\boldsymbol{\theta}_k).$$

Finally, $F_3(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k)$ is

$$F_3(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k)=\sum_{i=1}^{N}\sum_{j=0}^{q}p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k)\times \log\frac{p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k)}{p(\tau_i=j|\boldsymbol{\theta}_{k+1})},$$

while $F_3(\boldsymbol{\theta}_k|\boldsymbol{\theta}_k)$ is

$$F_3(\boldsymbol{\theta}_k|\boldsymbol{\theta}_k)=\sum_{i=1}^{N}\sum_{j=0}^{q}p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k)\times \log\frac{p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k)}{p(y(S_i)|\boldsymbol{\varphi}(S_i-\tau_i),\tau_i=j,\boldsymbol{\theta}_k)}.$$

According to the Kullback-Leibler divergence [26], we have

$$F_3(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k)\geqslant F_3(\boldsymbol{\theta}_k|\boldsymbol{\theta}_k).$$

Therefore, the following inequality holds,

$$F(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}_k)\geqslant F(\boldsymbol{\theta}_k|\boldsymbol{\theta}_k).$$

It demonstrates that the function $F(\boldsymbol{\theta})$ is monotonically increasing.

REFERENCES

[1] F. Giri and E.W. Bai, *Block-Oriented Nonlinear System Identification*, Springer, Berlin, 2010.

[2] T. Söderström and P. Stoica, *Systen Identification*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[3] M. Gan, G.Y. Chen, L. Chen, and C.L.P. Chen, "Term selection for a class of nonlinear separable models," *IEEE Trans. Neur. Net. Lear. Syst.*, vo. 31, no. 2, pp. 445-451, Feb. 2020.

[4] J. Chen, D.Q. Wang, Y.J. Liu, and Q.M. Zhu, "Varying infimum gradient descent algorithm for agent-sever systems using different order iterative preconditioning methods," *IEEE Trans. Ind. Inform.*, vol. 70, Apr. 2021. DOI: 10.1109/TII.2021.3123304

[5] C.P. Yu, L. Ljung, A. Wills, and M. Verhaegen, "Constrained subspace method for the identification of structured state-space models," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4201-4214, Oct. 2020.

[6] C.Y. Lai, G.D. Feng, K. Mukherjee, J. Tjong, and N.C. Kar, "Maximum torque per ampere control for IPMSM using gradient descent algorithm based on measured speed harmonics," *IEEE Trans. Ind. Inform.*, vol. 14, no. 4, pp. 1424-1435, Apr. 2018.

[7] J. Chen, B. Huang, F. Ding, and Y. Gu, "Variational Bayesian approach for ARX systems with missing observations and varying time-delays," *Automatica*, vol. 94, pp. 194-204, Aug. 2018.

[8] Y. Irshad, M. Mossberg, and T. Söderström, "System identification in a networked environment using second order statistical properties," *Automatica*, vol. 49, pp. 652-659, Feb. 2013.

[9] A. Padoan and A. Astolfi, "A note on delay coordinates for locally observable analytic systems," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1409-1412, May 2016.

[10] N. Sammaknejad, Y. Zhao, and B. Huang, "A review of the expectation maximization algorithm in data-driven process identification," *J. Process Control*, vol. 73, pp. 123-136, Jan. 2019.

[11] R.B. Gopaluni, "A particle filter approach to identification of nonlinear process under missing obsevations," *Can. J. Chem. Eng.*, vol. 86, no. 6, pp. 1081-1092, Dec. 2008.

[12] L. Xie, H.Z. Yang, and B. Huang, "FIR model identification of multirate processes with random delays using EM algorithm," *AIChE J.*, vol. 59, no. 11, pp. 4124-4132, May 2013.

[13] Y.J. Lu, B. Huang, and S. Khatibisepehr, "A variational Bayesian approach to robust identification of switched ARX models," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2542-2547, Dec. 2016.

[14] D.Q. Wang, S. Zhang, M. Gan, and J.L. Qiu, "A novel EM identification method for Hammerstein systems with missing output data," *IEEE Trans. Ind. Inform.*, vol. 16, no. 4, pp. 2500-2508, Apr. 2020.

[15] X.Q. Yang, X. Liu, and Z. Li, "Multimodel approach to robust identification of multiple-Input single-output nonlinear time-delay systems," *IEEE Trans. Ind. Inform.*, vol. 16, no. 4, pp. 2413-2422, Apr. 2020.

[16] Y. Zhao, A. Fatehi, and B. Huang, "A data-driven hybrid ARX and Markov-Chain modeling approach to process identification with time varying time delays," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4226-4236, May 2017.

[17] Y. Zhao, A. Fatehi, and B. Huang, "Robust estimation of ARX models with time varying time delays using variational Bayesian," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 532-542, Feb. 2018.

[18] J.X. Ma, J. Chen, W.L. Xiong, and F. Ding, "Expectation maximization estimation algorithm for Hammerstein models with non-Gaussian noise and random time delay from dual-rate sampled-data," *Digit. Signal Process.*, vol. 73, pp. 135-144, Feb. 2018.

[19] E.W. Bai, "Identification of linear systems with hard input nonlinearities of known structure," *Automatica*, vol. 38, no. 5, pp. 853-860, May 2002.

[20] G.C. Goodwin and K.S. Sin, *Adaptive Filtering, Prediction and Control*, Englewood CliPs, NJ: Prentice-Hall, 1984.

[21] S. Magnusson, C. Enyioha, et al., "Convergence of limited communication gradient methods," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1356-1371, May 2018.

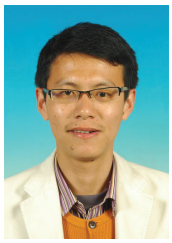
[22] F. Ding, G. Liu, and X.P. Liu, "Partially coupled stochastic gradient identification methods for non-uniformly sampled systems," *IEEE Trans. Autom. Control*, vol. 55, no. 8, pp. 1976-1981, Aug. 2010.

[23] Y. Saad and M.H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Comput.*, vol. 7, no. 3, pp. 856-869, Jul. 1986.

[24] F. Ding, "Coupled-least-squares identification for multivariable systems," *IET Control Theory Appl.*, vol. 7, no. 1, pp. 68-79, Jan. 2013.

[25] Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, 2003.

[26] L. Aggoune, Y. Chetouani, and T. Raïsi, "Fault detection in the distillation column process using Kullback Leibler divergence," *ISA Trans.*, vol. 63, pp. 394-400, Jul. 2016.



Jing Chen received his B.Sc. degree in the School of Mathematical Science and M.Sc. degree in the School of Information Engineering from Yangzhou University (Yanghzou, China) in 2003 and 2006, respectively, and received his Ph.D. degree in the School of Internet of Things Engineering, Jiangnan University (Wuxi, China) in 2013. He is currently a Professor in the School of Science, Jiangnan University (Wuxi, China). He is a Colleges and Universities Blue Project Middle-Aged Academic Leader (Jiangsu, China). His research interests include processing control and system identification.

Jie Gao received her B.Sc. degree in the Department of Mathematics from Nanjing Normal University (Nanjing, China) in 1994, M.Sc. degree in the Department of Mathematics from Nanjing University (Nanjing, China) in 2003, and Ph.D. degree in the School of Information Engineering, Jiangnan University (Wuxi, China) in 2009. She is currently a Professor in the School of Science, Jiangnan University (Wuxi, China). She is a Leader of Key Disciplines (Mathematics) in Jiangsu Province during the 14th Five-Year Plan period. Her research interests include probability and statistics.

Yanjun Liu received the B.Sc. degree from Jiangsu University of Technology (Changzhou, China) in 2003, the M.Sc. degree and the Ph.D. degree from Jiangnan University (Wuxi, China) in 2009 and 2012, respectively. She is currently an associate Professor in the School of Internet of Things Engineering, Jiangnan University. Her research interests are system identification and parameter estimation.

Cheng Wang received the B.Sc. degree from Beijing Jiaotong University (Beijing, China) in 2006 and the Ph.D. degree from Beijing Jiaotong University (Beijing, China) in 2014. He is currently an associate Professor in the School of Internet of Things Engineering, Jiangnan University. His research interests are system identification, data mining and computer vision.

Quanmin Zhu is Professor in control systems at the Department of Engineering Design and Mathematics, University of the West of England, Bristol, UK. He obtained his MSc in Harbin Institute of Technology, China in 1983 and PhD in Faculty of Engineering, University of Warwick, UK in 1989. His main research interest is in the area of nonlinear system modelling, identification, and control. He has published over 250 papers on these topics, edited various books with Springer, Elsevier, and the other publishers, and provided consultancy to various industries. Currently, Professor Zhu is acting as Editor of International Journal of Modelling, Identification and Control, Editor of International Journal of Computer Applications in Technology, Academic Editor of Complexity, Hindawi, Member of various journal editorial boards, and Editor of Elsevier book series of Emerging Methodologies and Applications in Modelling, Identification and Control. He is the founder and president of a series annual International Conference on Modelling, Identification and Control.