Signal Processing 140 (2017) 97–103

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Short communication

A multi-innovation state and parameter estimation algorithm for a state space system with d-step state-delay☆

Ling Xu[a], Feng Ding[a,b,*], Ya Gu[a], Ahmed Alsaedi[b], Tasawar Hayat[b,c]

[a] Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, PR China
[b] Nonlinear Analysis and Applied Mathematics (NAAM) Research Group, Department of Mathematics, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[c] Department of Mathematics, Quaid-I-Azam University, Islamabad 44000, Pakistan

ARTICLE INFO

Article history:
Received 7 May 2016
Revised 21 April 2017
Accepted 4 May 2017
Available online 5 May 2017

Keywords:
Signal filtering
Parameter estimation
Multi-innovation theory
State space system

ABSTRACT

This paper considers the state and parameter estimation problem of a state-delay system. On the basis of the stochastic gradient algorithm (i.e., the gradient based search estimation algorithm), this work extends the scalar innovation into an innovation vector and presents a multi-innovation gradient parameter estimation algorithm for a state-space system with d-step state-delay by means of the multi-innovation identification theory. For thesystems whose states are unknown, we use the states of the state observer for the parameter estimation and use the estimated parameters for the state estimation. This forms a joint multi-innovation state and parameter estimation algorithm for the state-delay systems with immeasurable states. The simulation results indicate that the proposed algorithms can work well.

1. Introduction

Parameter estimation is a central issue of system identification [1,2], controller design [3–6] and signal filtering [7,8]. Mathematical models are the essential characteristics of a system and system identification uses the statistical methods to build the mathematical models of dynamical systems from measured data [2]. The typical parameter estimation algorithms include the recursive methods and the iterative methods. They have widely applications in finding the roots of equations or solving matrix equations and in implementing the parameter estimation algorithms [9]. Xu et al. studied parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration [10] and presented a damping iterative parameter identification method for dynamical systems based on the sine signal measurement [11]. Chen et al. proposed a hierarchical gradient parameter estimation algorithm for Hammerstein nonlinear systems using the key term separation principle [12]. Xu presented a Newton iteration algorithm for the parameter estimation of dynamical systems [13].

There exist many identification methods for linear systems [14–16] and nonlinear systems [17–19], e.g., the generalized projection algorithm [20], the auxiliary model based interval-varying recursive least squares algorithm [21] and the decomposition based least squares iterative algorithm [22]. In the area of state space system identification, a modified subspace identification method was proposed for periodically non-uniformly sampled systems by using the lifting technique [23]. A state space least mean square filtering and an extended state space recursive least squares filtering were proposed for state space systems without considering input signals [24,25]. A state filtering based recursive least squares parameter estimation algorithm was proposed for linear systems with d-step state-delay [26]. Also, Wang et al.proposed a recursive parameter and state estimation algorithm for an input nonlinear state space system using the hierarchical identification principle [27]. Pan et al.discussed the image noise smoothing using a modified Kalman filter [28] and Wan et al presented a T-wave alternans assessment method based on least squares curve fitting technique [29].

In engineering, the states of some systems are immeasurable. Consequently, the state estimation of a system plays a significant role in control design and system identification. There exist many state and parameter estimation methods. In this literature, Guo and Zhao studied the convergence of an extended state observer for nonlinear systems with uncertainty [30]; Zhu presented the state estimation and unknown input reconstruction via both reduced-order and high-order sliding mode observers [31]. This paper uses

☆ This work was supported by the Science Research of Colleges Universities in Jiangsu Province (No. 16KJB120007, China), the National Natural Science Foundation of China (61273194) and the 111 Project (B12018).

* Corresponding author.
E-mail address: fding@jiangnan.edu.cn (F. Ding).

http://dx.doi.org/10.1016/j.sigpro.2017.05.006
0165-1684/

the state observer to generate the estimates of the states for a dynamic system from the available input-output data. Other methods include the auxiliary model based least squares algorithm for a dual-rate state space system with time-delay using the data filtering [32] and the state filtering based least squares iterative parameter estimation for observer canonical state space systems using decomposition [33].

The time-delay systems have drawn a great deal of attention of many researchers in the field of system control and system analysis [34–36]. For example, Gao et al. studied an artificial bee colony algorithm with space contraction for unknown parameters identification and time-delays of chaotic systems [37]; Li et al. discussed the local discontinuous Galerkin method for reaction-diffusion dynamical systems with time-delays [38]. In the previous work, a state filtering based recursive least squares parameter estimation algorithm [26] and a stochastic gradient (SG) parameter estimation algorithm [39] have been proposed for a state space system with d-step state-delay. On the basis of the work in [26,39], this paper uses the multi-innovation identification theory, extends the SG algorithm and derives a joint multi-innovation SG (MISG) parameter estimation and state estimation algorithm, where the parameter estimation uses the MISG algorithm and the state estimation uses the parameter estimates based observer. The methods in this paper can simultaneously estimate the parameters and states of the systems and differs from the previous methods which were based on the hierarchical identification principle [40].

This paper is organized as follows. Section 2 describes the canonical state space systems with state-delay. Section 3 derives the parameter identification model of the systems. Section 4 presents a combined multi-innovation stochastic gradient (MISG) parameter and state estimation algorithm and a state observer based recursive least squares parameter estimation algorithm for comparing the computational complexity. Section 5 provides an illustrative example. Finally, we offer some concluding remarks in Section 6.

2. The canonical state space model for state-delay systems

Let us introduce some notations. "$A =: X$" or "$X := A$" stands for "A is defined as X"; the symbol $\boldsymbol{I}$ ($\boldsymbol{I}_n$) stands for an identity matrix of appropriate size ($n \times n$); z represents a unit forward shift operator: $z\boldsymbol{x}(k) = \boldsymbol{x}(k+1)$ and $z^{-1}\boldsymbol{x}(k) = \boldsymbol{x}(k-1)$; the superscript T denotes the matrix/vector transpose; $\hat{\boldsymbol{\rho}}(k)$ denotes the estimate of $\boldsymbol{\rho}$ at time k; $\mathbf{1}_n$ represents an $n \times 1$ vector whose elements are all unity; the norm of a matrix $\boldsymbol{X}$ is defined as $\|\boldsymbol{X}\|^2 := \mathrm{tr}[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}]$.

Consider the following state space system with d-step state-delay,

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{x}(k-d) + \boldsymbol{f}u(k), \tag{1}$$

$$y(k) = \boldsymbol{c}\boldsymbol{x}(k) + v(k), \tag{2}$$

$$\boldsymbol{A} := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ a_n & a_{n-1} & a_{n-2} & \cdots & a_1 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

$$\boldsymbol{B} := \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_n \end{bmatrix} \in \mathbb{R}^{n\times n}, \ \boldsymbol{b}_i \in \mathbb{R}^{1\times n}, \quad \boldsymbol{f} := \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \in \mathbb{R}^{n},$$

$$\boldsymbol{c} := [1, 0, 0, \ldots, 0] \in \mathbb{R}^{1\times n},$$

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}$ is the system input, $y(k) \in \mathbb{R}$ is the system output, $v(k) \in \mathbb{R}$ is a random noise with zero mean, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times n}$, $\boldsymbol{f} \in \mathbb{R}^n$ and $\boldsymbol{c} \in \mathbb{R}^{1\times n}$ are the system parameter matrices/vectors.

The system matrices/vector $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{f}$ are the unknown parameters to be estimated from the input-output data $\{u(k), y(k)$: $k = 1, 2, \cdots\}$, $\boldsymbol{x}(k)$ is immeasurable. Assume that $(\boldsymbol{c}, \boldsymbol{A})$ is observable and $u(k) = 0$ and $y(k) = 0$ for $k \le 0$. If we remove $\boldsymbol{B}\boldsymbol{x}(k-d)$, then Eq. (1) becomes a standard state-space model. In this work, we suppose that the delay $d > 0$ is a known integer.

The data are collected from the dynamical system in (1)–(2). In general, one uses an uncorrelated sequence as the input signal for parameter identification.

3. The identification model

This section derives the identification model of the canonical state space model in (1)–(2). From (1), we have

$$x_i(k+1) = x_{i+1}(k) + \boldsymbol{b}_i\boldsymbol{x}(k-d) + f_i u(k), \ \ i = 1, 2, \ldots, n-1, \tag{3}$$

$$\begin{aligned} x_n(k+1) = {} & a_n x_1(k) + a_{n-1}x_2(k) + \cdots + a_1 x_n(k) \\ & + \boldsymbol{b}_n\boldsymbol{x}(k-d) + f_n u(k). \end{aligned} \tag{4}$$

Let $\boldsymbol{a} := [a_n, a_{n-1}, \ldots, a_1]^{\mathrm{T}} \in \mathbb{R}^n$. Using the properties of the shift operator z, multiplying Eqs. (3) by z^{-i} and (4) by z^{-n}, and adding all expressions give

$$\begin{aligned} x_1(k) = {} & \boldsymbol{a}\boldsymbol{x}(k-n) + \boldsymbol{b}_1\boldsymbol{x}(k-d-1) + \boldsymbol{b}_2\boldsymbol{x}(k-d-2) \\ & + \cdots + \boldsymbol{b}_{n-1}\boldsymbol{x}(k-d-n+1) + \boldsymbol{b}_n\boldsymbol{x}(k-d-n) \\ & + f_1 u(k-1) + f_2 u(k-2) + \cdots + f_n u(k-n). \end{aligned} \tag{5}$$

When $d \leqslant n-1$, define the information vector $\boldsymbol{\varphi}(k)$ and the parameter vector $\boldsymbol{\rho}$:

$$\begin{aligned} \boldsymbol{\varphi}(k) := {} & [\boldsymbol{x}^{\mathrm{T}}(k-d-1), \boldsymbol{x}^{\mathrm{T}}(k-d-2), \ldots, \boldsymbol{x}^{\mathrm{T}}(k-n), \ldots, \\ & \boldsymbol{x}^{\mathrm{T}}(k-n-d), u(k-1), u(k-2), \ldots, u(k-n)]^{\mathrm{T}} \in \mathbb{R}^{n^2+n}, \\ \boldsymbol{\rho} := {} & [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{a} + \boldsymbol{b}_{n-d}, \ldots, \boldsymbol{b}_n, \boldsymbol{f}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{n^2+n}. \end{aligned}$$

When $d \ge n$, define the information vector $\boldsymbol{\varphi}(k)$ and the parameter vector $\boldsymbol{\rho}$:

$$\begin{aligned} \boldsymbol{\varphi}(k) := {} & [\boldsymbol{x}^{\mathrm{T}}(k-n), \boldsymbol{x}^{\mathrm{T}}(k-d-1), \boldsymbol{x}^{\mathrm{T}}(k-d-2), \ldots, \\ & \boldsymbol{x}^{\mathrm{T}}(k-d-n+1), \boldsymbol{x}^{\mathrm{T}}(k-d-n), \\ & u(k-1), u(k-2), \ldots, u(k-n)]^{\mathrm{T}} \in \mathbb{R}^{n^2+2n}, \\ \boldsymbol{\rho} := {} & [\boldsymbol{a}, \boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n, \boldsymbol{f}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{n^2+2n}. \end{aligned}$$

From (2) and (5), we have

$$y(k) = x_1(k) + v(k) = \boldsymbol{\varphi}^{\mathrm{T}}(k)\boldsymbol{\rho} + v(k). \tag{6}$$

Remark 1. Eq. (6) is the identification model of the state space system with d-step state-delay. The information vector $\boldsymbol{\varphi}(k)$ contains the system input $u(t-i)$ and the unknown state vector $\boldsymbol{x}(k-i)$ and $\boldsymbol{\rho}$ contains all the parameters of the system.

Example 1. For a 2-state space system with d-step state-delay,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_2 & a_1 \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}\boldsymbol{x}(k-d) + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}u(k),$$

$$y(k) = [1, \ 0]\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + v(k).$$

Expanding this matrix equation gives

$$x_1(k+1) = x_2(k) + \boldsymbol{b}_1\boldsymbol{x}(k-d) + f_1 u(k), \tag{7}$$

$$x_2(k+1) = a_2x_1(k) + a_1x_2(k) + \boldsymbol{b}_2\boldsymbol{x}(k-d) + f_2u(k). \tag{8}$$

Multiplying Eqs. (7) by z^{-1} and (8) by z^{-2} gives

$$z^{-1}x_1(k+1) = z^{-1}x_2(k) + z^{-1}\boldsymbol{b}_1\boldsymbol{x}(k-d) + z^{-1}f_1u(k),$$
$$z^{-2}x_2(k+1) = z^{-2}a_2x_1(k) + z^{-2}a_1x_2(k) + z^{-2}\boldsymbol{b}_2\boldsymbol{x}(k-d) + z^{-2}f_2u(k),$$

or

$$x_1(k) = x_2(k-1) + \boldsymbol{b}_1\boldsymbol{x}(k-d-1) + f_1u(k-1),$$
$$x_2(k-1) = a_2x_1(k-2) + a_1x_2(k-2) + \boldsymbol{b}_2\boldsymbol{x}(k-d-2) + f_2u(k-2).$$

Adding the above two equations gives

$$\begin{aligned} x_1(k) &= a_2x_1(k-2) + a_1x_2(k-2) + \boldsymbol{b}_1\boldsymbol{x}(k-d-1) \\ &\quad + \boldsymbol{b}_2\boldsymbol{x}(k-d-2) + f_1u(k-1) + f_2u(k-2) \\ &= \boldsymbol{a}\boldsymbol{x}(k-2) + \boldsymbol{b}_1\boldsymbol{x}(k-d-1) + \boldsymbol{b}_2\boldsymbol{x}(k-d-2) \\ &\quad + f_1u(k-1) + f_2u(k-2). \end{aligned}$$

Its corresponding input-output representation is given by

$$\begin{aligned} y(k) &= \boldsymbol{a}\boldsymbol{x}(k-2) + \boldsymbol{b}_1\boldsymbol{x}(k-d-1) + \boldsymbol{b}_2\boldsymbol{x}(k-d-2) \\ &\quad + f_1u(k-1) + f_2u(k-2) + v(k). \end{aligned}$$

When $d = 1$, we have

$$\begin{aligned} x_1(k) &= a_2x_1(k-2) + a_1x_2(k-2) + \boldsymbol{b}_1\boldsymbol{x}(k-2) + \boldsymbol{b}_2\boldsymbol{x}(k-3) \\ &\quad + f_1u(k-1) + f_2u(k-2) \\ &= (\boldsymbol{a} + \boldsymbol{b}_1)\boldsymbol{x}(k-2) + \boldsymbol{b}_2\boldsymbol{x}(k-3) + f_1u(k-1) + f_2u(k-2). \end{aligned}$$

Hence, we have the input-state-output representation:

$$\begin{aligned} y(k) &= (\boldsymbol{a} + \boldsymbol{b}_1)\boldsymbol{x}(k-2) + \boldsymbol{b}_2\boldsymbol{x}(k-3) \\ &\quad + f_1u(k-1) + f_2u(k-2) + v(k). \end{aligned}$$

4. The parameter and state estimation algorithm

This section derives a multi-innovation stochastic gradient algorithm to estimate the parameter vector $\boldsymbol{\rho}$ in (6) and uses the observer to estimate the state vector $\boldsymbol{x}(k+1)$ of the system.

4.1. The SG algorithm

According to the identification model in (6), defining and minimizing the cost function,

$$J_1(\boldsymbol{\rho}) := \frac{1}{2}[y(k) - \boldsymbol{\varphi}^{\mathrm{T}}(k)\boldsymbol{\rho}]^2,$$

and using the gradient search, we may obtain a stochastic gradient algorithm:

$$\hat{\boldsymbol{\rho}}(k) = \hat{\boldsymbol{\rho}}(k-1) + \frac{\boldsymbol{\varphi}(k)}{r(k)}[y(k) - \boldsymbol{\varphi}^{\mathrm{T}}(k)\hat{\boldsymbol{\rho}}(k-1)], \tag{9}$$

$$r(k) = r(k-1) + \|\boldsymbol{\varphi}(k)\|^2, \ r(0) = 1. \tag{10}$$

Here, $1/r(k)$ is the step-size or convergence factor. The choice of $r(k)$ guarantees that the parameter estimation error converges to zero. However, difficulties arise in that the information vector $\boldsymbol{\varphi}(k)$ contains the unknown state vector $\boldsymbol{x}(k-i)$ $(i = 1+d, 2+d, \ldots, n+d)$, the SG algorithm in (9)–(10) cannot compute the estimate $\hat{\boldsymbol{\rho}}(k)$ of $\boldsymbol{\rho}$. The approach here is to replace the unknown state vector $\boldsymbol{x}(k-i)$ in $\boldsymbol{\varphi}(k)$ with its $\hat{\boldsymbol{x}}(k-i)$. The unknown vector $\boldsymbol{\varphi}(k)$ in (9)–(10) with its estimate $\hat{\boldsymbol{\varphi}}(k)$ in (13) or (14), we can obtain the following stochastic gradient (SG) algorithm for estimating $\boldsymbol{\rho}$:

$$\hat{\boldsymbol{\rho}}(k) = \hat{\boldsymbol{\rho}}(k-1) + \frac{\hat{\boldsymbol{\varphi}}(k)}{r(k)}[y(k) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k)\hat{\boldsymbol{\rho}}(k-1)], \tag{11}$$

$$r(k) = r(k-1) + \|\hat{\boldsymbol{\varphi}}(k)\|^2, \ r(0) = 1, \tag{12}$$

$$\begin{aligned} \hat{\boldsymbol{\varphi}}(k) &= [\hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-2), \ldots, \hat{\boldsymbol{x}}^{\mathrm{T}}(k-n), \ldots, \\ &\quad \hat{\boldsymbol{x}}^{\mathrm{T}}(k-n-d), u(k-1), u(k-2), \ldots, u(k-n)]^{\mathrm{T}}, \\ &\quad d \leqslant n-1, \end{aligned} \tag{13}$$

$$\begin{aligned} \hat{\boldsymbol{\varphi}}(k) &= [\hat{\boldsymbol{x}}^{\mathrm{T}}(k-n), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-2), \ldots, \\ &\quad \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-n+1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-n), \\ &\quad u(k-1), u(k-2), \ldots, u(k-n)]^{\mathrm{T}}, \ d \geqslant n. \end{aligned} \tag{14}$$

The quantity $e(k) := y(k) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k)\hat{\boldsymbol{\rho}}(k-1) \in \mathbb{R}$ is called the innovation.

Remark 2. The SG algorithm for estimating $\boldsymbol{\rho}$ has slow convergence rates. This motivates us to study the MISG algorithm for improving the convergence rates in the following.

4.2. The MISG algorithm

In order to improve the accuracy of the SG algorithm, we extend the SG algorithm and derive a multi-innovation stochastic gradient algorithm by expanding the innovation length.

Define an innovation vector:

$$\boldsymbol{E}_{\rho}(p,k) := \begin{bmatrix} e(k) \\ e(k-1) \\ \vdots \\ e(k-p+1) \end{bmatrix} \in \mathbb{R}^p,$$

where the positive integer p represents the innovation length, and

$$e(k-i) = y(k-i) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k-i)\hat{\boldsymbol{\rho}}(k-i-1).$$

In general, one may think that the estimate $\hat{\boldsymbol{\rho}}(k-1)$ is closer to $\boldsymbol{\rho}$ than $\hat{\boldsymbol{\rho}}(k-i)$ at time $k-i$ $(i = 2, 3, 4, \ldots, p-1)$. Thus, we take the innovation vector as

$$\boldsymbol{E}_{\rho}(p,k) := \begin{bmatrix} y(k) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k)\hat{\boldsymbol{\rho}}(k-1) \\ y(k-1) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k-1)\hat{\boldsymbol{\rho}}(k-1) \\ \vdots \\ y(k-p+1) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k-p+1)\hat{\boldsymbol{\rho}}(k-1) \end{bmatrix} \in \mathbb{R}^p.$$

Define the stacked information matrix $\hat{\boldsymbol{\Phi}}(p,k)$ and the stacked output vector $\boldsymbol{Y}(p,k)$ as

$$\hat{\boldsymbol{\Phi}}(p,k) := [\hat{\boldsymbol{\varphi}}(k), \hat{\boldsymbol{\varphi}}(k-1), \ldots, \hat{\boldsymbol{\varphi}}(k-p+1)] \in \mathbb{R}^{n \times p},$$
$$\boldsymbol{Y}(p,k) := [y(k), y(k-1), \ldots, y(k-p+1)]^{\mathrm{T}} \in \mathbb{R}^p.$$

Then, the innovation vector $\boldsymbol{E}_{\rho}(p,k)$ can be equivalently expressed as

$$\boldsymbol{E}_{\rho}(p,k) = \boldsymbol{Y}(p,k) - \hat{\boldsymbol{\Phi}}^{\mathrm{T}}(p,k)\hat{\boldsymbol{\rho}}(k-1).$$

Therefore, replacing $y(k)$, $\hat{\boldsymbol{\varphi}}(k)$ and $e(k)$ in (11) with $\boldsymbol{Y}(p,k)$, $\hat{\boldsymbol{\Phi}}(p,k)$ and $\boldsymbol{E}_{\rho}(p,k)$, we can obtain the following multi-innovation stochastic gradient (MISG) algorithm with the innovation length p:

$$\hat{\boldsymbol{\rho}}(k) = \hat{\boldsymbol{\rho}}(k-1) + \frac{\hat{\boldsymbol{\Phi}}(p,k)}{r(k)}\boldsymbol{E}_{\rho}(p,k), \tag{15}$$

$$\boldsymbol{E}_{\rho}(p,k) = \boldsymbol{Y}(p,k) - \hat{\boldsymbol{\Phi}}^{\mathrm{T}}(p,k)\hat{\boldsymbol{\rho}}(k-1), \tag{16}$$

$$r(k) = r(k-1) + \|\hat{\boldsymbol{\varphi}}(k)\|^2,\ r(0) = 1, \tag{17}$$

$$\boldsymbol{Y}(p,k) = [y(k), y(k-1), \ldots, y(k-p+1)]^{\mathrm{T}}, \tag{18}$$

$$\hat{\boldsymbol{\Phi}}(p,k) = [\hat{\boldsymbol{\varphi}}(k), \hat{\boldsymbol{\varphi}}(k-1), \ldots, \hat{\boldsymbol{\varphi}}(k-p+1)], \tag{19}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}(k) = [&\hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-2), \ldots, \hat{\boldsymbol{x}}^{\mathrm{T}}(k-n), \ldots,\\ &\hat{\boldsymbol{x}}^{\mathrm{T}}(k-n-d), u(k-1), u(k-2), \ldots, u(k-n)]^{\mathrm{T}},\\ &\text{for } d \leqslant n-1,\end{aligned} \tag{20}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}(k) = [&\hat{\boldsymbol{x}}^{\mathrm{T}}(k-n), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-2), \ldots,\\ &\hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-n+1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-n),\\ &u(k-1), u(k-2), \ldots, u(k-n)]^{\mathrm{T}}, \quad \text{for } d \geqslant n.\end{aligned} \tag{21}$$

When the innovation length $p = 1$, the MISG algorithm degrades to the SG algorithm.

Remark 3. The innovation is the useful information which can improve the estimation accuracy. Thus, the MISG algorithm can provide more accurate parameter estimates by expanding the innovation length.

4.3. The state estimation algorithm

Using the parameter estimation vector $\hat{\boldsymbol{\rho}}(k)$ to form the system matrices $\hat{\boldsymbol{A}}(k)$, $\hat{\boldsymbol{B}}(k)$ and $\hat{\boldsymbol{f}}(k)$ and based on the canonical state space model in (1)–(2), we can use the following observer to estimate the state vector $\boldsymbol{x}(k)$:

$$\hat{\boldsymbol{x}}(k+1) = \hat{\boldsymbol{A}}(k)\hat{\boldsymbol{x}}(k) + \hat{\boldsymbol{B}}(k)\hat{\boldsymbol{x}}(k-d) + \hat{\boldsymbol{f}}(k)u(k), \tag{22}$$

$$\hat{\boldsymbol{A}}(k) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \ddots & \vdots\\ \vdots & \vdots & \ddots & \ddots & 0\\ 0 & 0 & \cdots & 0 & 1\\ \hat{a}_n(k) & \hat{a}_{n-1}(k) & \hat{a}_{n-2}(k) & \cdots & \hat{a}_1(k)\end{bmatrix}, \tag{23}$$

$$\hat{\boldsymbol{B}}(k) = \begin{bmatrix}\hat{\boldsymbol{b}}_1(k)\\ \hat{\boldsymbol{b}}_2(k)\\ \vdots\\ \hat{\boldsymbol{b}}_n(k)\end{bmatrix}, \quad \hat{\boldsymbol{f}}(k) = \begin{bmatrix}\hat{f}_1(k)\\ \hat{f}_2(k)\\ \vdots\\ \hat{f}_n(k)\end{bmatrix}, \tag{24}$$

$$\begin{aligned}\hat{\boldsymbol{\rho}}(k) = [&\hat{\boldsymbol{b}}_1(k), \hat{\boldsymbol{b}}_2(k), \ldots, \hat{\boldsymbol{a}}(k) + \hat{\boldsymbol{b}}_{n-d}(k), \ldots, \hat{\boldsymbol{b}}_n(k), \hat{\boldsymbol{f}}^{\mathrm{T}}(k)]^{\mathrm{T}},\\ &d \leqslant n-1,\end{aligned} \tag{25}$$

$$\hat{\boldsymbol{\rho}}(k) = [\hat{\boldsymbol{a}}(k), \hat{\boldsymbol{b}}_1(k), \hat{\boldsymbol{b}}_2(k), \ldots, \hat{\boldsymbol{b}}_{n-1}(k), \hat{\boldsymbol{b}}_n(k), \hat{\boldsymbol{f}}^{\mathrm{T}}(k)]^{\mathrm{T}},\ d \geqslant n. \tag{26}$$

The steps of computing the parameter estimate $\hat{\boldsymbol{\rho}}(k)$ in (15)–(21) and the state estimate $\hat{\boldsymbol{x}}(k+1)$ in (22)–(26) are listed in the following.

1. Let $k = 1$, set the initial values $\hat{\boldsymbol{\rho}}(0) = \mathbf{1}_{n^2+2n}/p_0$, $r(0) = 1$, $\hat{\boldsymbol{x}}(k-i) = \mathbf{1}_n/p_0$, $i = 0, 1, \ldots, d+n$, $p_0 = 10^6$. Give the innovation length p.

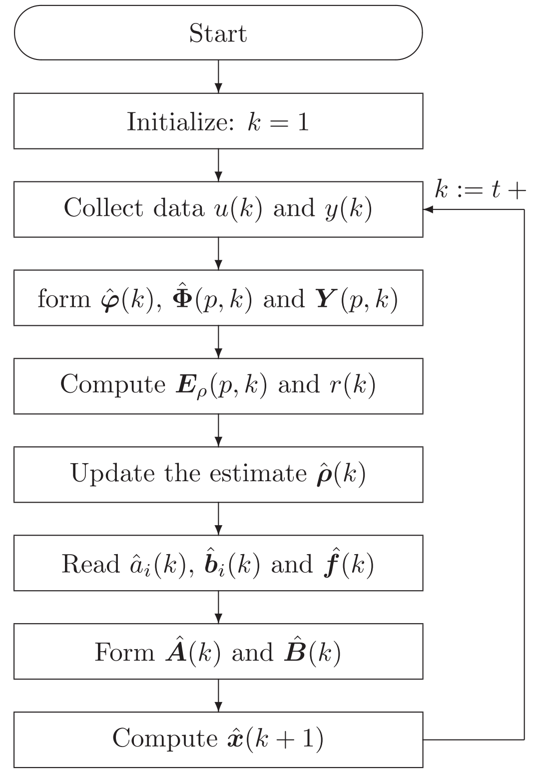

Fig. 1. The flowchart of computing the parameter estimate $\hat{\boldsymbol{\rho}}(k)$ and the state estimate $\hat{\boldsymbol{x}}(k+1)$.

2. Collect the input-output data $u(k)$ and $y(k)$ and form $\hat{\boldsymbol{\varphi}}(k)$ by (20) or (21), $\hat{\boldsymbol{\Phi}}(p,k)$ by (19) and $\boldsymbol{Y}(p, k)$ by (18).
3. Compute $\boldsymbol{E}_\rho(p, k)$ by (17) and $r(k)$ by (16).
4. Update the parameter estimation vector $\hat{\boldsymbol{\rho}}(k)$ by (15).
5. Read $\hat{a}_i(k)$, $\hat{\boldsymbol{b}}_i(k)$ and $\hat{\boldsymbol{f}}(k)$ from $\hat{\boldsymbol{\rho}}(k)$ according to the definition of $\hat{\boldsymbol{\rho}}(k)$.
6. Form $\hat{\boldsymbol{A}}(k)$ and $\hat{\boldsymbol{B}}(k)$ by (23) and (24).
7. Compute the state estimation vector $\hat{\boldsymbol{x}}(k+1)$ by (22).
8. Increase k by 1 and go to Step 2, continue the recursive calculation.

The flowchart of computing the parameter estimation vector $\hat{\boldsymbol{\rho}}(k)$ in (15)–(21) and the state estimate $\hat{\boldsymbol{x}}(k+1)$ in (22)–(26) are shown in Fig. 1.

Remark 4. Eqs. (15)–(21) and (22)–(26) form the joint parameter and state estimation algorithm for interactively computing the parameter estimate $\hat{\boldsymbol{\rho}}(k)$ and the state estimate $\hat{\boldsymbol{x}}(k+1)$, where the parameter estimation uses the MISG algorithm – see (15) to (21), and the state estimation uses the parameter estimates based observer – see (22) to (26).

4.4. The RLS algorithm for comparison

The following gives the recursive least squares (RLS) algorithm for comparison.

According to the identification model in (6), define the least squares cost function,

$$J_2(\boldsymbol{\rho}) := \sum_{j=1}^{t}[y(j) - \boldsymbol{\varphi}^{\mathrm{T}}(j)\boldsymbol{\rho}]^2,$$

and using the least squares principle, we can obtain the recursive relation stochastic gradient algorithm:

$$\hat{\boldsymbol{\rho}}(k) = \hat{\boldsymbol{\rho}}(k-1) + \boldsymbol{P}_\rho(k)\boldsymbol{\varphi}(k)[y(k) - \boldsymbol{\varphi}^{\mathrm{T}}(k)\hat{\boldsymbol{\rho}}(k-1)], \tag{27}$$

$$\boldsymbol{P}_{\rho}^{-1}(k) = \boldsymbol{P}_{\rho}^{-1}(k-1) + \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^{\mathrm{T}}(k),\ \ \boldsymbol{P}_{\rho}(0) = p_0\boldsymbol{I}_{n^2+n}. \tag{28}$$

Applying the matrix inversion lemma $(\boldsymbol{A}+\boldsymbol{BC})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{I} + \boldsymbol{CA}^{-1}\boldsymbol{B})^{-1}\boldsymbol{CA}^{-1}$ to (28) gives

$$\boldsymbol{P}_{\rho}(k) = \boldsymbol{P}_{\rho}(k-1) - \frac{\boldsymbol{P}_{\rho}(k-1)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^{\mathrm{T}}(k)\boldsymbol{P}_{\rho}(k-1)}{1+\boldsymbol{\varphi}^{\mathrm{T}}(k)\boldsymbol{P}_{\rho}(k-1)\boldsymbol{\varphi}(k)}. \tag{29}$$

Similarly, a difficulty arises because the information vector $\boldsymbol{\varphi}(k)$ contains the unknown state vector $\boldsymbol{x}(k-i)$ $(i = 1+d, 2+d, \ldots, n+d)$, the algorithm in (27) and (29) is impossible to generate $\hat{\boldsymbol{\rho}}(k)$ of $\boldsymbol{\rho}$. The approach here is to replace the unknown vector $\boldsymbol{\varphi}(k)$ in (27) and (29) with its estimate $\hat{\boldsymbol{\varphi}}(k)$ in (33) or (34), and define the gain vector

$$\boldsymbol{L}(k) := \boldsymbol{P}_{\rho}(k)\hat{\boldsymbol{\varphi}}(k) = \boldsymbol{P}_{\rho}(k-1)\hat{\boldsymbol{\varphi}}(k)[1+\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k)\boldsymbol{P}_{\rho}(k-1)\hat{\boldsymbol{\varphi}}(k)]^{-1},$$

we can obtain the following recursive least squares (RLS) algorithm for estimating $\boldsymbol{\rho}$:

$$\hat{\boldsymbol{\rho}}(k) = \hat{\boldsymbol{\rho}}(k-1) + \boldsymbol{L}(k)[y(k) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k)\hat{\boldsymbol{\rho}}(k-1)], \tag{30}$$

$$\boldsymbol{L}(k) = \boldsymbol{P}_{\rho}(k-1)\hat{\boldsymbol{\varphi}}(k)[1+\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k)\boldsymbol{P}_{\rho}(k-1)\hat{\boldsymbol{\varphi}}(k)]^{-1}, \tag{31}$$

$$\boldsymbol{P}_{\rho}(k) = \boldsymbol{P}_{\rho}(k-1) - \boldsymbol{L}(k)[\boldsymbol{P}_{\rho}(k-1)\hat{\boldsymbol{\varphi}}(k)]^{\mathrm{T}},\ \ \boldsymbol{P}_{\rho}(0) = p_0\boldsymbol{I}_{n^2+n}, \tag{32}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}(k) = [&\hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-2), \ldots, \hat{\boldsymbol{x}}^{\mathrm{T}}(k-n), \ldots,\\ &\hat{\boldsymbol{x}}^{\mathrm{T}}(k-n-d), u(k-1), u(k-2), \ldots,\\ &u(k-n)]^{\mathrm{T}},\ \ d \leqslant n-1,\end{aligned} \tag{33}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}(k) = [&\hat{\boldsymbol{x}}^{\mathrm{T}}(k-n), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-2), \ldots,\\ &\hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-n+1), \hat{\boldsymbol{x}}^{\mathrm{T}}(k-d-n), u(k-1), u(k-2), \ldots,\\ &u(k-n)]^{\mathrm{T}}, d \geqslant n.\end{aligned} \tag{34}$$

Remark 5. Eqs. (30)–(34) and (22)–(26) form the joint parameter and state estimation algorithm for interactively computing the parameter estimate $\hat{\boldsymbol{\rho}}(k)$ and the state estimate $\hat{\boldsymbol{x}}(k+1)$, where the parameter estimation uses the RLS algorithm – see (30) to (34), and the state estimation uses the parameter estimates based observer – see (22) to (26).

4.5. The comparison of the computational complexity

It has been just pointed out by Golub and Van Loan in [41] that the flop (floating point operation) counting is a necessarily crude approach to the measuring of program efficiency since it ignores subscripting, memory traffic, and the countless other overheads associated with program execution, the flop counting is just a "quick and dirty" accounting method that captures only one of the several dimensions of the efficiency issue although multiplication/division and addition/subtraction with different length are different [32].

The flop numbers of the MISG and RLS algorithms at each recursion are shown in Tables 1 and 2. Their total flops are, respectively, given by

$$N_1 := (4p+2)(n^2+n) + p,$$
$$N_2 := 4(n^2+n)^2 + 6(n^2+n).$$

In order to compare the computational efficiency, we count the difference between the amount of calculation of these two algorithms. When $n_a > 2$ and $n_b > 2$, $n_a n_b > n_a + n_b$, $N_2 > 4(n_a+n_b+n_c+n_d)^2 + 6(n_a+n_b+n_c+n_d)$. Then we have

$$\begin{aligned}N_2 - N_1 &> 4(n^2+n)^2 + 6(n^2+n) - [(4p+2)(n^2+n)+p]\\ &= 4(n^2+n-p)(n^2+n) + 4(n^2+n) - p > 0,\\ &\quad \text{as } p < n^2+n.\end{aligned}$$

From here, we can see that the MISG algorithm has less computational load than the RLS algorithm. For instance, when $n = p = 10$, we have $N_2 - N_1 = 49060 - 4630 = 44430$ flops because the RLS algorithm requires computing the covariance matrix $\boldsymbol{P}_{\rho}(k)$ of large size $(n^2+n) \times (n^2+n)$.

5. Example

The system should be stable. Thus, the parameter matrices in Example are selected so that all the variables of the system are bounded.

Consider the following state space system with 2-step state-delay:

$$\begin{aligned}\boldsymbol{x}(k+1) &= \begin{bmatrix} 0 & 1 \\ -0.01 & -0.22 \end{bmatrix}\boldsymbol{x}(k) + \begin{bmatrix} 0.19 & -0.08 \\ 0.16 & -0.12 \end{bmatrix}\boldsymbol{x}(k-2)\\ &\quad + \begin{bmatrix} 0.8 \\ -0.8 \end{bmatrix}u(k),\\ y(k) &= [1,\ 0]\boldsymbol{x}(k) + v(k).\end{aligned}$$

The parameter vector to be identified is

$$\begin{aligned}\boldsymbol{\rho} &= [a_2, a_1, b_{11}, b_{12}, b_{21}, b_{22}, f_1, f_2]^{\mathrm{T}}\\ &= [-0.01, -0.22, 0.19, -0.08, 0.16, -0.12, 0.8, -0.8]^{\mathrm{T}}.\end{aligned}$$

In simulation, the input $\{u(k)\}$ is taken as an uncorrelated persistent excitation signal sequence with zero mean and unit variance, and $\{v(k)\}$ as a white noise sequence with zero mean and variance $\sigma^2 = 0.30^2$. We apply the state estimation based MISG algorithm in (15)–(21) to estimate the parameters of this example system. The parameter estimates and their estimation errors are shown in Table 3 and the parameter estimation errors δ versus k are shown in Fig. 2 with $p = 1$, $p = 2$ and $p = 5$, respectively.

From Table 3 and Fig. 2, we can draw the following conclusions.

- The parameter estimates given by the MISG algorithm converge fast to their true values for large p – see Table 3.
- The MISG algorithm with $p \geq 2$ has higher accuracy than the SG algorithm – see Fig. 2.
- The parameter estimation errors given by the MISG algorithm become smaller with the data length k and the innovation length p increasing – see Table 3 and Fig. 2.
- The proposed algorithm can estimate effectively the parameters and states of a class of time-delay state space system under the stochastic framework.

6. Conclusions

Based on the multi-innovation theory, this paper extends the stochastic gradient algorithm and presents a multi-innovation stochastic gradient algorithm for state space systems with d-step state-delay, where the unknown states are estimated using an state observer and the unknown state vector in the parameter estimation algorithm are replaced with the corresponding estimated states of the observer. Furthermore, a state observer based recursive least squares parameter estimation algorithm is given for comparison. The theoretical analysis and simulation results indicate

Table 1
The computational efficiency of the MISG algorithm.

Variables	Numbers of multiplications	Numbers of additions
$\hat{\boldsymbol{\rho}}(k)=\hat{\boldsymbol{\rho}}(k-1)+\hat{\boldsymbol{\Phi}}(p,k)[\boldsymbol{E}_\rho(p,k)/r(k)]\in\mathbb{R}^{n^2+n}$	$p(n^2+n)+p$	$p(n^2+n)$
$\boldsymbol{E}_\rho(p,k)=\boldsymbol{Y}(p,k)-\hat{\boldsymbol{\Phi}}^{\mathrm{T}}(p,k)\hat{\boldsymbol{\rho}}(k-1)\in\mathbb{R}^p$	$p(n^2+n)$	$p(n^2+n)$
$r(k)=r(k-1)+\|\hat{\boldsymbol{\varphi}}(k)\|^2\in\mathbb{R}$	n^2+n	n^2+n
Sum	$(2p+1)(n^2+n)+p$	$(2p+1)(n^2+n)$
Total flops	$N_1:=(4p+2)(n^2+n)+p$	

Table 2
The computational efficiency of the RLS algorithm.

Variables	Numbers of multiplications	Numbers of additions
$\hat{\boldsymbol{\rho}}(k)=\hat{\boldsymbol{\rho}}(k-1)+\boldsymbol{L}(k)e(k)\in\mathbb{R}^{n^2+n}$	n^2+n	n^2+n
$e(k)=y(k)-\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k)\hat{\boldsymbol{\rho}}(k-1)\in\mathbb{R}$	n^2+n	n^2+n
$\boldsymbol{L}(k)=\boldsymbol{\zeta}(k)/[1+\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(k)\boldsymbol{\zeta}(k)]\in\mathbb{R}^{n^2+n}$	$2(n^2+n)$	n^2+n
$\boldsymbol{\zeta}(k):=\boldsymbol{P}_\rho(k-1)\hat{\boldsymbol{\varphi}}(k)\in\mathbb{R}^{n^2+n}$	$(n^2+n)^2$	$(n^2+n)^2-(n^2+n)$
$\boldsymbol{P}_\rho(k)=\boldsymbol{P}_\rho(k-1)-\boldsymbol{L}(k)\hat{\boldsymbol{\zeta}}^{\mathrm{T}}(k)\in\mathbb{R}^{n^2+n}$	$(n^2+n)^2$	$(n^2+n)^2$
Sum	$2(n^2+n)^2+4(n^2+n)$	$2(n^2+n)^2+2(n^2+n)$
Total flops	$N_2:=4(n^2+n)^2+6(n^2+n)$	

Table 3
The parameter estimates and errors with $\sigma^2=0.30^2$.

Algorithms	k	a_2	a_1	b_{11}	b_{12}	b_{21}	b_{22}	f_1	f_2	δ (%)
SG	100	−0.00912	−0.15036	0.04307	−0.06630	0.02505	−0.01004	0.49833	−0.38674	47.52683
(MISG, $p=1$)	200	−0.00071	−0.15897	0.03430	−0.06226	0.03583	−0.01874	0.51905	−0.41671	44.53744
	500	−0.01158	−0.16218	0.04174	−0.07458	0.04926	−0.02578	0.55028	−0.45958	39.90966
	1000	−0.01300	−0.16742	0.04407	−0.07779	0.05693	−0.02935	0.57120	−0.48809	36.94100
	2000	−0.01608	−0.17243	0.04797	−0.08160	0.06551	−0.03294	0.58943	−0.51204	34.32511
	3000	−0.01644	−0.17616	0.04817	−0.08312	0.07199	−0.03621	0.59949	−0.52557	32.86166
MISG, $p=2$	100	0.00359	−0.10678	0.07097	−0.11681	0.07671	−0.04656	0.66219	−0.54089	29.99962
	200	0.01599	−0.12109	0.06185	−0.11177	0.09003	−0.05590	0.68106	−0.57759	26.68912
	500	0.00850	−0.13309	0.07485	−0.12515	0.10632	−0.06220	0.70645	−0.62350	22.16532
	1000	0.01018	−0.14260	0.07880	−0.12629	0.11409	−0.06384	0.72115	−0.65182	19.62474
	2000	0.00966	−0.15129	0.08537	−0.12916	0.12141	−0.06584	0.73406	−0.67410	17.47593
	3000	0.01029	−0.15688	0.08764	−0.13042	0.12688	−0.06787	0.74038	−0.68567	16.37969
MISG, $p=5$	100	−0.01604	−0.16063	0.16648	−0.04942	0.12274	−0.11018	0.77804	−0.72833	9.28037
	200	−0.00941	−0.17261	0.16361	−0.05183	0.13136	−0.11273	0.78161	−0.75681	6.94835
	500	−0.01370	−0.18867	0.17725	−0.06884	0.14888	−0.11953	0.79311	−0.77446	3.86359
	1000	−0.00956	−0.19552	0.17681	−0.06720	0.15297	−0.11685	0.79300	−0.78499	3.00103
	2000	−0.00865	−0.20173	0.18160	−0.07054	0.15557	−0.11647	0.79656	−0.79077	2.10426
	3000	−0.00843	−0.20539	0.18430	−0.07311	0.15762	−0.11675	0.79793	−0.79282	1.61476
True values		−0.01000	−0.22000	0.19000	−0.08000	0.16000	−0.12000	0.80000	−0.80000	

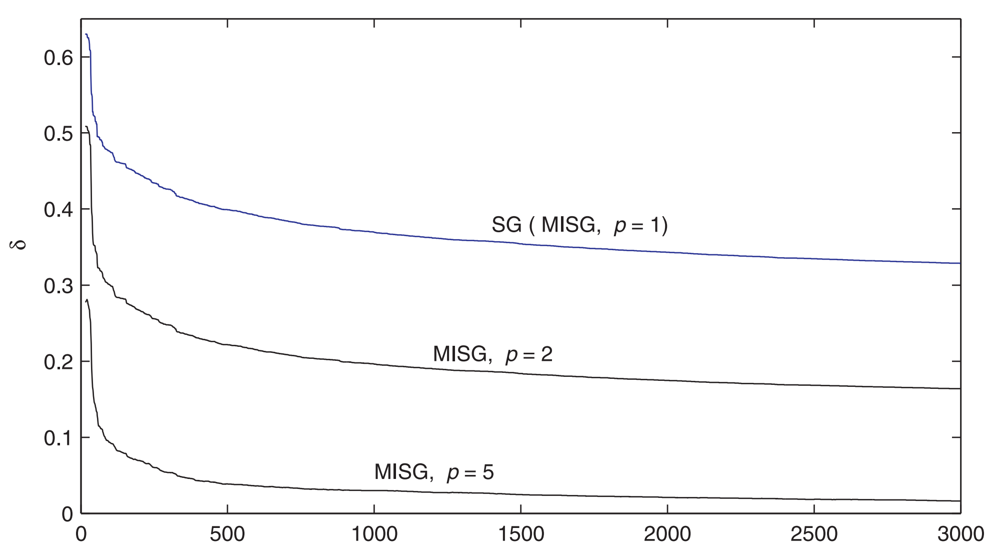

Fig. 2. The parameter estimation errors δ versus k with $\sigma^2=0.30^2$.

that the proposed algorithm can generate highly accurate parameter estimates for large innovation lengths and the computational burden becomes large but this increased computation is still tolerable and affordable for modern computers. The methods in this paper can be extended to study the state and parameter estimation problems of uncertain chaotic delayed nonlinear state space systems and hybrid switching-impulsive dynamical network systems [42] and applied to other fields [43–45].

References

[1] G.C. Goodwin, K.S. Sin, Adaptive Filtering Prediction and Control, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
[2] L. Ljung, System Identification: Theory for the User, 2nd ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1999.
[3] H. Li, W.S. Yan, Y. Shi, Continuous-time model predictive control of under-actuated spacecraft with bounded control torques, Automatica 75 (2016) 144–153.
[4] H. Li, Y. Shi, W. Yan, On neighbor information utilization in distributed receding horizon control for consensus-seeking, IEEE Trans. Cybern. 46 (9) (2016) 2019–2027.
[5] H. Li, Y. Shi, W. Yan, Distributed receding horizon control of constrained nonlinear vehicle formations with guaranteed γ-gain stability, Automatica 68 (2016) 148–154.
[6] X.J. Su, L.G. Wu, P. Shi, Y.D. Song, A novel approach to output feedback control of fuzzy stochastic systems, Automatica 50 (12) (2014) 3268–3275.
[7] X.J. Su, P. Shi, L.G. Wu, Y.D. Song, Fault detection filtering for nonlinear switched stochastic systems, IEEE Trans. Automat. Control 61 (5) (2016) 1310–1315.
[8] X.J. Su, P. Shi, L.G. Wu, M.V. Basin, Reliable filtering with strict dissipativity for t-s fuzzy time-delay systems, IEEE Trans. Cybern. 44 (12) (2014) 2470–2483.
[9] D.Q. Wang, W. Zhang, Improved least squares identification algorithm for multivariable Hammerstein systems, J. Franklin Inst. 352 (11) (2015) 5292–5370.
[10] L. Xu, L. Chen, W.L. Xiong, Parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration, Nonlinear Dyn. 79 (3) (2015) 2155–2163.
[11] L. Xu, The damping iterative parameter identification method for dynamical systems based on the sine signal measurement, Signal Process. 120 (2016) 660–667.
[12] H.B. Chen, Y.S. Xiao, Hierarchical gradient parameter estimation algorithm for Hammerstein nonlinear systems using the key term separation principle, Appl. Math. Comput. 247 (2014) 1202–1210.
[13] L. Xu, Application of the Newton iteration algorithm to the parameter estimation for dynamical systems, J. Comput. Appl. Math. 288 (2015) 33–43.
[14] Y.J. Wang, F. Ding, Novel data filtering based parameter identification for multiple-input multiple-output systems using the auxiliary model, Automatica 71 (2016) 308–313.
[15] Y.J. Wang, F. Ding, The filtering based iterative identification for multivariable systems, IET Control Theory Appl. 10 (8) (2016) 894–902.
[16] Y.J. Wang, F. Ding, The auxiliary model based hierarchical gradient algorithms and convergence analysis using the filtering technique, Signal Process. 128 (2016) 212–221.
[17] D.Q. Wang, Hierarchical parameter estimation for a class of MIMO Hammerstein systems based on the reframed models, Appl. Math. Lett. 57 (2016) 13–19.
[18] Y.J. Wang, F. Ding, Recursive least squares algorithm and gradient algorithm for Hammerstein–Wiener systems using the data filtering, Nonlinear Dyn. 84 (2) (2016) 1045–1053.
[19] F. Ding, X.M. Liu, M.M. Liu, The recursive least squares identification algorithm for a class of Wiener nonlinear systems, J. Franklin Inst. 353 (7) (2016) 1518–1526.
[20] F. Ding, L. Xu, Q.M. Zhu, Performance analysis of the generalised projection identification for time-varying systems, IET Control Theory Appl. 10 (18) (2016) 2506–2514.
[21] F. Ding, F.F. Wang, L. Xu, T. Hayat, A. Alsaedi, Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique, IET Control Theory Appl. 11 (3) (2017) 390–400.
[22] F. Ding, F.F. Wang, L. Xu, M.H. Wu, Decomposition based least squares iterative identification algorithm for multivariate pseudo-linear ARMA systems using the data filtering, J. Franklin Inst. 354 (3) (2017) 1321–1339.
[23] J. Ding, J.X. Lin, Modified subspace identification for periodically non-uniformly sampled systems by using the lifting technique, Circuits Syst. Signal Process. 33 (5) (2014) 1439–1449.
[24] M.B. Malik, M. Salman, State-space least mean square, Digital Signal Process. 18 (3) (2008) 334–345.
[25] A. Irshad, M. Salman, S. Bashir, M.B. Malik, Extended state space recursive least squares, Digital Signal Process. 49 (2016) 95–103.
[26] Y. Gu, F. Ding, J.H. Li, State filtering and parameter estimation for linear systems with d-step state-delay, IET Signal Process. 8 (6) (2014) 639–646.
[27] X.H. Wang, F. Ding, Recursive parameter and state estimation for an input nonlinear state space system using the hierarchical identification principle, Signal Process. 117 (2015) 208–218.
[28] J. Pan, X.H. Yang, H.F. Cai, B.X. Mu, Image noise smoothing using a modified Kalman filter, Neurocomputing 173 (2016) 1625–1629.
[29] X.K. Wan, Y. Li, C. Xia, M.H. Wu, J. Liang, N. Wang, A T-wave alternans assessment method based on least squares curve fitting technique, Measurement 86 (2016) 93–100.
[30] B.Z. Guo, Z.L. Zhao, On the convergence of an extended state observer for nonlinear systems with uncertainty, Syst. Control Lett. 60 (6) (2011) 420–430.
[31] F.L. Zhu, State estimation and unknown input reconstruction via both reduced-order and high-order sliding mode observers, J. Process Control 22 (1) (2012) 296–302.
[32] F. Ding, X.M. Liu, Y. Gu, An auxiliary model based least squares algorithm for a dual-rate state space system with time-delay using the data filtering, J. Franklin Inst. 353 (2) (2016) 398–408.
[33] F. Ding, X.M. Liu, X.Y. Ma, Kalman state filtering based least squares iterative parameter estimation for observer canonical state space systems using decomposition, J. Comput. Appl. Math. 301 (2016) 135–143.
[34] L. Xu, A proportional differential control method for a time-delay system using the taylor expansion approximation, Appl. Math. Comput. 236 (2014) 391–399.
[35] C. Wang, Y. Shen, Improved delay-dependent robust stability criteria for uncertain time delay systems, Appl. Math. Comput. 218 (6) (2011) 2880–2888.
[36] X.Y. Li, S.L. Sun, H-infinity filtering for networked linear systems with multiple packet dropouts and random delays, Digital Signal Process. 46 (2015) 59–67.
[37] F. Gao, F.X. Fei, A novel artificial bee colony algorithm with space contraction for unknown parameters identification and time-delays of chaotic systems, Appl. Math. Comput. 219 (2) (2012) 552–568.
[38] D.F. Li, C.J. Zhang, H.Y. Qin, LDG method for reaction-diffusion dynamical systems with time delay, Appl. Math. Comput. 217 (22) (2011) 9173–9181.
[39] Y. Gu, R. Ding, Stochastic gradient algorithm for state space system with d-step delay, in: The 2014 26th Chinese Control and Decision Conference, May 31 – June 2, Changsha, China, 2014, pp. 1924–1928.
[40] Y. Gu, F. Ding, J.H. Li, States based iterative parameter estimation for a state space model with multi-state delays using decomposition, Signal Process. 106 (2015) 294–300.
[41] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[42] Y. Ji, X.M. Liu, Unified synchronization criteria for hybrid switching-impulsive dynamical networks, Circuits Syst. Signal Process. 34 (5) (2015) 1499–1517.
[43] L. Feng, M.H. Wu, Q.X. Li, Array factor forming for image reconstruction of one-dimensional nonuniform aperture synthesis radiometers, IEEE Geosci. Remote Sens. Lett. 13 (2) (2016) 237–241.
[44] T.Z. Wang, J. Qi, H. Xu, Fault diagnosis method based on FFT-RPCA-SVM for cascaded-multilevel inverter, ISA Trans. 60 (2016) 156–163.
[45] T.Z. Wang, H. Wu, M.Q. Ni, An adaptive confidence limit for periodic non-steady conditions fault detection, Mech. Syst. Signal Process. 72–73 (2016) 328–345.