Robust standard gradient descent algorithm for ARX models using Aitken acceleration technique

Jing Chen, Min Gan, Quanmin Zhu, Narayan Pritesh, Yanjun Liu

***Abstract*—A robust standard gradient descent algorithm for ARX models using Aitken acceleration method is developed. Considering that the standard gradient descent algorithm has slow convergence rates and is sensitive to the step-size, a robust and accelerative standard gradient descent algorithm is derived. This algorithm is based on Aitken acceleration method, and its convergence rate is improved from linear convergence to at least quadratic convergence in general. Furthermore, the robust and accelerative standard gradient descent algorithm is always convergent with no limitation of the step-size. Both the convergence analysis and the simulation examples demonstrate that the presented algorithm is effective.**

***Index Terms*—Parameter estimation, standard gradient descent algorithm, Aitken acceleration technique, convergence rate, ARX model**

I. Introduction

System identification plays an important role in control engineering, for the reason that robust controller designs often need the parameters of the systems to be known in prior [1]–[5]. Generally, two directions are involved in the system identification: model structure identification and parameter estimation [6]–[8]. Model structure identification is the base and more challenging; while parameter estimation has the assumption that the model structure of the system is known, and then the parameters are estimated by using some identification algorithms. These algorithms roughly include the standard gradient descent (SGD) algorithm [9], the least squares algorithm [10], [11], the iterative algorithm and the expectation-maximization algorithm [12], [13]. Among these algorithms, the SGD algorithm does not require to solve for the matrix's inverse, thus has less computational efforts [14]. However, gradient descent is relatively slow close to the minimum: technically, its asymptotic rate of convergence is inferior to many other methods. For poorly conditioned convex problems, gradient descent increasingly 'zigzags' as the gradients point nearly orthogonally to the shortest direction to a minimum point. In addition, the SGD algorithm is sensitive to the step-size: a small step-size leads to slow convergence rates, while a large one makes the algorithm divergent.

J. Chen is with the School of Science, Jiangnan University, Wuxi 214122, PR China (chenjing1981929@126.com)

M. Gan is with the College of Computer Science & Technology, Qingdao University, Qingdao, PR China (aganmin@gmail.com)

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

Y.J. Liu is with Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, PR China (yanjunliu_1983@126.com)

This work is supported by the National Natural Science Foundation of China (Nos. 61973137, 62073082), the Natural Science Foundation of Jiangsu Province (No. BK20201339), the Fundamental Research Funds for the Central Universities (No. JUSRP22016) and the Funds of the Science and Technology on Near-Surface Detection Laboratory (No. TCGZ2019A001)

In order to increase the convergence rate of the SGD algorithm, some modified gradient descent algorithms are developed over the past few decades, e.g., changing the direction of the gradient descent to get an optimal one, or computing a suitable step-size in each iteration [15], [16]. For example, Abbasbandy et al provided a conjugate gradient method for fuzzy symmetric positive definite system of linear equations [17], in which the conjugate gradient method can obtain an optimal direction in each iteration. To get a suitable step-size, Ma et al proposed a forgetting factor gradient descent algorithm for Hammerstein systems with saturation and preload nonlinearities by using data filtering method [18]; Chen et al derived a modified gradient descent algorithm for ARX models by introducing a convergence index in the step-size, and then the convergence rates of the gradient descent algorithm are increased [19]. Although these two kinds of algorithms can increase the convergence rates, they also bring some issues, e.g., a big oscillation when the parameter estimates are close to the true values, or heavy computational efforts when computing a better direction and a suitable step-size in each iteration.

Recently, a multi-step-length gradient iterative (MUL-GI) algorithm is developed to increase the convergence rates in a new way, and its basic idea is to assign a direction and a corresponding step-size for each element in the parameter vector, where the columns in the information matrix are independent [20]. The MUL-GI algorithm can obtain the best parameter estimates in one iteration and is robust to the initial parameter values. However, the information vector must be turned into a new information vector by using the Gram-Schmidt orthogonalization method in the MUL-GI algorithm, which will increase the computational efforts. For machine learning, two outstanding modified stochastic gradient algorithms are developed: one is the stochastic average gradient (SAG) algorithm [21], and the other is the stochastic variance reduced gradient (SVRG) algorithm [22]. Both these two algorithms can increase the convergence rates from sub-linear to linear and have less computational efforts, with a prerequisite that the step-size needs to meet certain conditions.

The Aitken technique is an accelerative method which is usually used for solving matrix equations. Its main idea is to apply a transformed iterative function to replace the original unchanging iterative function, and then the convergence rate will be improved [23], [24]. For example, in [25], an Aitken-Newton iterative method for nonlinear equations was developed, and the method is better than certain optimal methods of same convergence order. In [26], an improved Aitken acceleration method for solving nonlinear equations was presented, which can get the solutions of the nonlinear equations quickly. In system identification, Wang et al proposed an Aitken-based stochastic gradient (SG) algorithm for ARX models with time delay [27]. Since the SG algorithm is an on-line algorithm whose iterative function is changed at each sampling instant, it is doubtful in terms of the feasibility and effectiveness of the proposed procedure. As mentioned above, the slow convergence rate and the step-size calculation are the two major disadvantages of the SGD algorithm. To efficiently utilize the SGD algorithm to complex problems such as large-scale system identification or neural network learning, there remains a need for accurately integrating the Aitken acceleration technique into a comprehensive SGD framework for achieving a

much faster convergence rate and making the algorithm robust to the step-size. In this paper, a robust and accelerative SGD (RA-SGD) algorithm based on the Aitken technique is developed, which sets the following aims:

(1) The proposed algorithm can increase the convergence rate of the SGD algorithm from linear convergence to at least quadratic convergence.

(2) The proposed algorithm does not involve the step-size calculation, thus has less computational efforts, especially for large-scale systems.

(3) The proposed algorithm can make a divergent SGD algorithm convergent with no limitation of the step-size (is robust to the step-size).

Briefly, this paper is organized as follows. Section II introduces the ARX models and the SGD algorithm. Section III develops the RA-SGD algorithm. The convergence analysis is given in Section IV. Section V provides the simulation examples. Finally, Section VI sums up the paper and gives future directions.

II. The SGD algorithm for ARX models

Let us introduce some notations first. The symbol $\mathbf{I}$ stands for an identity matrix of the appropriate sizes; the norm of a matrix $\mathbf{X}$ is defined as $\|\mathbf{X}\| = \sqrt{\lambda_{\max}[\mathbf{X}\mathbf{X}^{\mathrm{T}}]}$; $\lambda_{\max}[\mathbf{X}\mathbf{X}^{\mathrm{T}}]$ means the maximum eigenvalue of matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$; the norm of a vector $\mathbf{z} = [z_1, z_2, \cdots, z_n]^{\mathrm{T}} \in \mathbb{R}^n$ is defined as $\|\mathbf{z}\| = (\sum_{i=1}^{n} |z_i|^2)^{\frac{1}{2}}$; the superscript T denotes the matrix transpose.

Consider an ARX model

$$A(z)y(t) = B(z)u(t) + v(t), \tag{1}$$

where $u(t)$ and $y(t)$ are the input and output, respectively, $\{u(t)\}$ is taken as a persistent excitation signal sequence, and $v(t)$ is a stochastic white noise with zero mean and variance σ^2, the polynomials $A(z)$ and $B(z)$ are expressed as

$$\begin{aligned} A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}, \\ B(z) &= b_1 + b_2 z^{-1} + \cdots + b_n z^{-n+1}. \end{aligned}$$

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

$$\begin{aligned} \boldsymbol{\theta} &= [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n]^{\mathrm{T}} \in \mathbb{R}^{2n}, \\ \boldsymbol{\varphi}(t) &= [-y(t-1), -y(t-2), \cdots, -y(t-n), u(t), \\ &\quad u(t-1), \cdots, u(t-n+1)]^{\mathrm{T}} \in \mathbb{R}^{2n}. \end{aligned}$$

Then the ARX model can be written by

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta} + v(t).$$

Collect L ($L \gg 2n$) input and output data and define

$$\begin{aligned} Y(L) &= [y(1), y(2), \cdots, y(L)]^{\mathrm{T}} \in \mathbb{R}^{L}, \\ \boldsymbol{\Phi}(L) &= [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \cdots, \boldsymbol{\varphi}(L)] \in \mathbb{R}^{2n \times L}, \\ V(L) &= [v(1), v(2), \cdots, v(L)]^{\mathrm{T}} \in \mathbb{R}^{L}. \end{aligned}$$

We can rewrite the ARX model in a similar form

$$Y(L) = \boldsymbol{\Phi}^{\mathrm{T}}(L)\boldsymbol{\theta} + V(L). \tag{2}$$

Minimizing the cost function

$$J(\boldsymbol{\theta}) = \frac{1}{2}[Y(L) - \boldsymbol{\Phi}^{\mathrm{T}}(L)\boldsymbol{\theta}]^{\mathrm{T}}[Y(L) - \boldsymbol{\Phi}^{\mathrm{T}}(L)\boldsymbol{\theta}]$$

gives the following standard gradient descent (SGD) algorithm

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \gamma\boldsymbol{\Phi}(L)[Y(L) - \boldsymbol{\Phi}^{\mathrm{T}}(L)\hat{\boldsymbol{\theta}}_{k-1}], \tag{3}$$

where γ is the step-size.

The choice of the step-size plays an important role in the SGD algorithm. Because a poor choice of the step-size in the SGD algorithm may lead to a slow convergence rate or even divergent results, e.g., a small step-size will lead to slow convergence rates, while a large one may lead to divergence.

Remark 1. In the light of the literature [28]–[30], the step-size should be chosen in $\left(0, \frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}\right)$; otherwise, the SGD algorithm will be divergent. The detailed derivation is given in [30] and hence omitted.

III. Robust and accelerative SGD algorithm

To improve the convergence rate of the SGD algorithm from linear convergence to at least quadratic convergence, a new SGD algorithm termed as robust and accelerative SGD (RA-SGD) algorithm is developed based on the Aitken technique.

Lemma 1 Aitken acceleration method [31]: Assume that the sequence $\{x_k\}$ is generated by the iterative function $\psi(x)$, that is

$$x_k = \psi(x_{k-1}),$$

let x_* be the limit of the sequence $\{x_k\}$, and

$$\lim_{k\to\infty} \frac{x_{k+1} - x_*}{x_k - x_*} = c, \ c \neq 1.$$

Then the sequence $\{\bar{x}_k\}$ generated by

$$\bar{x}_k = x_k - \frac{(x_{k+1} - x_k)^2}{x_{k+2} + x_k - 2x_{k+1}}$$

also converges to x_*, and its convergence rate is quicker than that of the sequence $\{x_k\}$.

Definition 1 [32]: Assume that the sequence $\{x_k\}_{k=0}^{\infty}$ converges to x_*, and let $e_k = x_k - x_*$. If

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = c,$$

where p ($p \geqslant 1$) and $c \neq 0$ are two constants, then the sequence $\{x_k\}$ is pth-order convergent. When $p = 1$, the sequence is linearly convergent; when $p = 2$, the sequence is quadratically convergent.

Lemma 2: Assume that x_* is a fixed point of the iterative function $\psi(x)$, the differential function $\psi'(x)$ is continuous in the neighborhood $(x_* - \xi, x_* + \xi)$ of point x_*, $\xi > 0$ is a constant, and $0 < |\psi'(x_*)| < 1$. Then the sequence $\{x_k\}$ generated by the iterative function

$$x_{k+1} = \psi(x_k)$$

is linearly convergent.

(The detailed derivation is given in Appendix A.)

Let $\boldsymbol{\theta}_*$ be the true parameters. For the SGD algorithm in (3), assume that the parameter estimates $\hat{\boldsymbol{\theta}}_k$ converge to the true values with the increase of k, then we have

$$\begin{aligned} \lim_{k\to\infty} \hat{\boldsymbol{\theta}}_k &= \lim_{k\to\infty} \hat{\boldsymbol{\theta}}_{k-1} + \gamma\boldsymbol{\Phi}(L)[Y(L) - \boldsymbol{\Phi}^{\mathrm{T}}(L) \lim_{k\to\infty} \hat{\boldsymbol{\theta}}_{k-1}], \\ \boldsymbol{\theta}_* &= \boldsymbol{\theta}_* + \gamma\boldsymbol{\Phi}(L)[Y(L) - \boldsymbol{\Phi}^{\mathrm{T}}(L)\boldsymbol{\theta}_*]. \end{aligned}$$

Replacing $\boldsymbol{\theta}_*$ with $\boldsymbol{\theta}$ yields

$$\psi(\boldsymbol{\theta}) = \boldsymbol{\theta} + \gamma\boldsymbol{\Phi}(L)[Y(L) - \boldsymbol{\Phi}^{\mathrm{T}}(L)\boldsymbol{\theta}], \tag{4}$$

where $\psi(\boldsymbol{\theta})$ is an iterative function.

Theorem 1: For the ARX model in (2), the true parameters are $\boldsymbol{\theta}_*$. The corresponding SGD algorithm is expressed by (3), the step-size $\gamma \in (0, \frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]})$. Then the SGD algorithm is linearly convergent.

Proof: Assume that the true parameter values are $\boldsymbol{\theta}_*$, we have

$$\mathbf{e}_{k+1} = \hat{\boldsymbol{\theta}}_{k+1} - \boldsymbol{\theta}_* = \psi(\hat{\boldsymbol{\theta}}_k) - \psi(\boldsymbol{\theta}_*) = \psi'(\boldsymbol{\varsigma})(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_*) = \psi'(\boldsymbol{\varsigma})\mathbf{e}_k.$$

Because

$$\psi'(\boldsymbol{\theta}) = \mathbf{I} - \gamma[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)].$$

In order to make sure that the SGD algorithm is convergent, the step-size must satisfy

$$\|\gamma[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]\| < 2.$$

It follows that the step-size γ should be chosen as

$$\gamma < \frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}.$$

Then

$$0 < \|\psi'(\boldsymbol{\theta})\| = \|\mathbf{I} - \gamma[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]]\| < 1.$$

From Lemma 2, we can conclude that the SGD algorithm is linearly convergent when $\gamma \in \left(0, \frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}\right)$. ■

For the reason that the SGD algorithm is linearly convergent, according to Lemma 1 and Definition 1, we can use the Aitken technique to accelerate the SGD algorithm.

Assume that the parameter estimates $\hat{\boldsymbol{\theta}}_k$ converge to the true parameters $\boldsymbol{\theta}_*$ and satisfy

$$\lim_{k\to\infty}[\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_*] \approx c[\hat{\boldsymbol{\theta}}_{k-1} - \boldsymbol{\theta}_*], \tag{5}$$

where c is a constant. Based on the Aitken technique, we have

$$[\hat{\boldsymbol{\theta}}_k + \hat{\boldsymbol{\theta}}_{k-2} - 2\hat{\boldsymbol{\theta}}_{k-1}]^{\mathrm{T}}\boldsymbol{\theta}_* \approx \hat{\boldsymbol{\theta}}_k^{\mathrm{T}}\hat{\boldsymbol{\theta}}_{k-2} - \hat{\boldsymbol{\theta}}_{k-1}^{\mathrm{T}}\hat{\boldsymbol{\theta}}_{k-1}. \tag{6}$$

For the considered ARX model, $\boldsymbol{\theta}$ is a vector, thus the optimal parameter vector estimate $\boldsymbol{\theta}_*$ cannot be computed by the above equation(one equation contains $2n$ unknown variables). To remedy this problem, we assume that each element in those parameter vectors satisfies the above equation. Define

$$\boldsymbol{\theta}_* = [\theta_*^1, \theta_*^2, \cdots, \theta_*^{2n}]^{\mathrm{T}},$$
$$\hat{\boldsymbol{\theta}}_k = [\hat{\theta}_k^1, \hat{\theta}_k^2, \cdots, \hat{\theta}_k^{2n}]^{\mathrm{T}}.$$

Equation (6) is transformed into the following $2n$ equations,

$$[\hat{\theta}_k^j + \hat{\theta}_{k-2}^j - 2\hat{\theta}_{k-1}^j]\theta_*^j \approx \hat{\theta}_k^j\hat{\theta}_{k-2}^j - \hat{\theta}_{k-1}^j\hat{\theta}_{k-1}^j, \quad j = 1, \cdots, 2n, \tag{7}$$

and each equation can be transformed into

$$\theta_*^j \approx \hat{\theta}_{k-2}^j - \frac{(\hat{\theta}_{k-1}^j - \hat{\theta}_{k-2}^j)^2}{\hat{\theta}_k^j + \hat{\theta}_{k-2}^j - 2\hat{\theta}_{k-1}^j}, \quad j = 1, \cdots, 2n. \tag{8}$$

It follows that the iterative function is written by

$$\bar{\theta}_k^j = \hat{\theta}_k^j - \frac{(\hat{\theta}_{k+1}^j - \hat{\theta}_k^j)^2}{\hat{\theta}_{k+2}^j + \hat{\theta}_k^j - 2\hat{\theta}_{k+1}^j}. \tag{9}$$

In summary, we can get the robust and accelerative SGD (RA-SGD) algorithm as follows:

$$\bar{a}_{k-2}^j = \hat{a}_{k-2}^j - \frac{(\hat{a}_{k-1}^j - \hat{a}_{k-2}^j)^2}{\hat{a}_k^j + \hat{a}_{k-2}^j - 2\hat{a}_{k-1}^j}, \quad j = 1, \cdots, n, \ k \geqslant 3, \tag{10}$$

$$\bar{b}_{k-2}^j = \hat{b}_{k-2}^j - \frac{(\hat{b}_{k-1}^j - \hat{b}_{k-2}^j)^2}{\hat{b}_k^j + \hat{b}_{k-2}^j - 2\hat{b}_{k-1}^j}, \quad j = 1, \cdots, n, \tag{11}$$

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \gamma\boldsymbol{\Phi}(L)[Y(L) - \boldsymbol{\Phi}^{\mathrm{T}}(L)\hat{\boldsymbol{\theta}}_{k-1}], \tag{12}$$

$$\hat{\boldsymbol{\theta}}_k = [\hat{a}_k^1, \cdots, \hat{a}_k^n, \hat{b}_k^1, \cdots, \hat{b}_k^n]^{\mathrm{T}}, \tag{13}$$

$$\bar{\boldsymbol{\theta}}_{k-2} = [\bar{a}_{k-2}^1, \cdots, \bar{a}_{k-2}^n, \bar{b}_{k-2}^1, \cdots, \bar{b}_{k-2}^n]^{\mathrm{T}}, \tag{14}$$

$$Y(L) = [y(1), y(2), \cdots, y(L)]^{\mathrm{T}}, \tag{15}$$

$$\boldsymbol{\Phi}(L) = [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \cdots, \boldsymbol{\varphi}(L)], \tag{16}$$

$$\boldsymbol{\varphi}(t) = [-y(t-1), \cdots, -y(t-n), u(t), \cdots, u(t-n+1)]^{\mathrm{T}}, \quad t = 1, \cdots, L, \tag{17}$$

$$0 < \gamma < \frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}. \tag{18}$$

The steps of computing the parameter estimation vector by using the RA-SGD algorithm are listed in the following.

1) Let $\hat{\boldsymbol{\theta}}_0 = \mathbf{1}/p_0$ and $\bar{\boldsymbol{\theta}}_0 = \mathbf{1}/p_0$ with $\mathbf{1}$ being a column vector whose entries are all unity and $p_0 = 10^6$.
2) Let $k = 1$, $y(j) = 0, u(j) = 0$, $j \leqslant 0$, and give a small positive number δ.
3) Collect all the input-output data $\{u(1), y(1)\}, \cdots, \{u(L), y(L)\}$.
4) Form $\boldsymbol{\varphi}(1), \cdots, \boldsymbol{\varphi}(L)$ by (17).
5) Form $Y(L)$ and $\boldsymbol{\Phi}(L)$ by (15) and (16), respectively.
6) Choose the step-size γ according to (18).
7) Update the parameter estimation vector $\hat{\boldsymbol{\theta}}_k$ by (12).
8) Compare k with 2, if $k \leqslant 2$, let $k = k + 1$ and go back to step 7; otherwise, go to the next step.
9) Compute each parameter estimate $\bar{a}_{k-2}^j$ and $\bar{b}_{k-2}^j$, $j = 1, \cdots, n$ by (10) and (11), respectively.
10) Form $\bar{\boldsymbol{\theta}}_{k-2}$ by (14).
11) Compare $\bar{\boldsymbol{\theta}}_{k-2}$ and $\bar{\boldsymbol{\theta}}_{k-3}$: if $\|\bar{\boldsymbol{\theta}}_{k-2} - \bar{\boldsymbol{\theta}}_{k-3}\| \leqslant \delta$, then terminate the procedure and obtain the $\bar{\boldsymbol{\theta}}_{k-2}$; otherwise, increase k by 1 and go to step 7.

Remark 2. Although the RA-SGD algorithm enjoys a faster convergence rate (at least quadratic convergence) than the SGD algorithm (linear convergence), the parameter estimates in some iterations may be abnormal. The reason is that the value of the denominator in (9) sometimes nearly equals to zero, i.e., $|\hat{\theta}_{k+2}^j + \hat{\theta}_k^j - 2\hat{\theta}_{k+1}^j|$ is very small, but $|\hat{\theta}_{k+1}^j - \hat{\theta}_k^j|$ is not.

Remark 3. The abnormal parameter estimate of the RA-SGD algorithm is mainly caused by the rounding error of the computer. However, the estimates quickly become normal, as shown in Fig. 3: the estimate in iteration 25 is abnormal, but the estimate in the next iteration approaches the true values. See Theorem 2 in Section IV.

IV. Convergence analysis

In this section, we compare the convergence rate of the RA-SGD algorithm with that of the SGD algorithm, and explain why the RA-SGD algorithm is robust to the step-size. Furthermore, the using range of the Aitken acceleration technique is also given.

Lemma 3 [31]: Assume that x_* is a fixed point of the iterative function $\psi(x)$, the pth-order differential function $\psi^{(p)}(x)$ is continuous in the neighborhood $(x_* - \xi, x_* + \xi)$ of point x_*, $p \geqslant 2$ is an integer, $\xi > 0$ is a constant, and the differential functions satisfy

$$\psi^{(l)}(x_*) = 0, \quad (l = 1, 2, \cdots, p-1), \quad \psi^{(p)}(x_*) \neq 0.$$

Then the errors $e_k = x_k - x_*$ satisfy

$$\lim_{k\to\infty}\frac{e_{k+1}}{e_k^p} = \frac{\psi^{(p)}(x_*)}{p!},$$

and the sequence $\{x_k\}$ generated by the iterative function

$$x_{k+1} = \psi(x_k)$$

is pth-order convergent.

Define an iterative function as

$$x_k = \psi(x_{k-1}),$$

and then the new iterative function obtained by using the Aitken acceleration technique is

$$\bar{x}_k = \phi(\bar{x}_{k-1}),$$

where

$$\phi(x) = x - \frac{[\psi(x) - x]^2}{\psi(\psi(x)) - 2\psi(x) + x}. \tag{19}$$

In order to get the convergence rate of the RA-SGD algorithm, the following lemma is presented.

Lemma 4: Assume that the iterative function $\psi(x)$ and its differential function $\psi'(x)$ are both continuous in the neighborhood $(x_*-\xi, x_*+\xi)$ of point x_*, $\xi>0$ is a constant, $\psi'(x_*)\neq 1$, and the iterative function $\phi(x)$ is expressed by Equation (19). Then x_* is a fixed point of $\psi(x)$ if and only if x_* is a fixed point of $\phi(x)$.

(The detailed derivation is given in Appendix B.)

Theorem 2: For the ARX model in (2), the corresponding RA-SGD algorithm is expressed by (10)-(17), and $\boldsymbol{\theta}_*$ is a fixed point of the following iterative function

$$\psi(\boldsymbol{\theta})=\boldsymbol{\theta}+\gamma\boldsymbol{\Phi}(L)[Y(L)-\boldsymbol{\Phi}^{\mathrm{T}}(L)\boldsymbol{\theta}],$$

where $Y(L)$, $\boldsymbol{\Phi}(L)$, and $\gamma>0$ keep unchanged during all the iterations. Then the RA-SGD algorithm is at least quadratically convergent.

(The proof is given in Appendix C.)

Remark 4. If $\boldsymbol{\theta}_*$ is a fixed point of the iterative function $\psi(\boldsymbol{\theta})$, then Theorem 2 illustrates that the parameter estimates of the RA-SGD algorithm always converge to the fixed point $\boldsymbol{\theta}_*$ even though the step-size $\gamma\geqslant\frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$. Thus, in the RA-SGD algorithm, we can choose a random positive constant for γ instead of computing it by Equation (18).

In conclusion, we have the following findings for different step-sizes.

- When the step-size in the SGD algorithm is suggested to be chosen equal to $\frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$, the iterative function is written by

  $$\psi(\boldsymbol{\theta})=\boldsymbol{\theta}-\frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}\boldsymbol{\Phi}(L)[Y(L)-\boldsymbol{\Phi}^{\mathrm{T}}(L)\boldsymbol{\theta}].$$

  Then

  $$\|\psi'(\boldsymbol{\theta}_*)\|=1.$$

  For this iterative function, we cannot guarantee that the parameter estimates of the SGD algorithm converge to the true values, which means that $\gamma=\frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$ is the threshold for convergence. Thus the conservative choice of γ for the SGD algorithm is $\gamma\in\left(0,\frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}\right)$.
- When the step-size $\gamma>\frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$, the SGD algorithm is divergent. Using the Aitken acceleration technique for this SGD algorithm yields

  $$\phi(\hat{\vartheta}_k^j)=\hat{\vartheta}_k^j-\frac{(\psi_j(\hat{\vartheta}_k^j)-\hat{\vartheta}_k^j)^2}{\psi_j(\psi_j(\hat{\vartheta}_k^j))+\hat{\vartheta}_k^j-2\psi_j(\hat{\vartheta}_k^j)}.$$

  and

  $$\psi'(\vartheta_*)\neq 1.$$

  According to Theorem 2, we can get that the divergent SGD algorithm becomes convergent by integrating the Aitken acceleration technique into it.
- When the step-size $\gamma=\frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$, one cannot guarantee the convergence of the SGD algorithm. However, we have

  $$\psi'(\vartheta_*)\neq 1,$$

  which means that the RA-SGD algorithm is convergent. This is also verified by Fig. 2 and Table III.

The convergence of the SGD and RA-SGD algorithms with different step-sizes is listed in Table I.

Remark 5. From Table I, we can get that the RA-SGD algorithm is always convergent when the step-size $\gamma>0$, which means that the RA-SGD algorithm is robust to the step-size.

Based on Lemma 4 and Theorem 2, we can conclude that when an ARX model contains hidden variables (e.g., missing outputs, varying time-delays), the RA-SGD algorithm would be invalid. That is to say, the Aitken technique cannot accelerate the convergence rates of the SGD algorithm when the systems have hidden variables. Taking the ARX model for example, when the model contains unknown outputs, the iterative function is

$$\hat{\boldsymbol{\theta}}_k=\hat{\boldsymbol{\theta}}_{k-1}+\gamma\hat{\boldsymbol{\Phi}}_{k-1}(L)[Y(L)-\hat{\boldsymbol{\Phi}}_{k-1}^{\mathrm{T}}(L)\hat{\boldsymbol{\theta}}_{k-1}],$$

in which

$$\begin{aligned}\hat{\boldsymbol{\Phi}}_{k-1}(L)&=[\hat{\boldsymbol{\varphi}}_{k-1}(1),\hat{\boldsymbol{\varphi}}_{k-1}(2),\cdots,\hat{\boldsymbol{\varphi}}_{k-1}(L)],\\ \hat{\boldsymbol{\varphi}}_{k-1}(t)&=[-\hat{y}_{k-1}(t-1),-\hat{y}_{k-1}(t-2),\cdots,-\hat{y}_{k-1}(t-n),\\ &\quad u(t),u(t-1),\cdots,u(t-n+1)]^{\mathrm{T}},t=1,\cdots,L,\\ \hat{y}_{k-1}(t-j)&=\hat{\boldsymbol{\varphi}}_{k-1}^{\mathrm{T}}(t-j)\hat{\boldsymbol{\theta}}_{k-1},\ j=1,\cdots,n.\end{aligned}$$

In this case, the iterative function is changed in each iteration for the reason that $\hat{\boldsymbol{\Phi}}_{k-1}(L)$ is varying in each iteration, thus according to Theorem 2, the Aitken based method is invalid.

Remark 6. When using the Aitken technique to accelerate the convergence rates of the algorithms, the iterative functions of these algorithms should keep unchanging; otherwise, the Aitken acceleration technique would be invalid. For this reason, the algorithm proposed in [27] needs to be further discussed.

Remark 7. This study shows that the RA-SGD algorithm achieves a much faster convergence rate and is robust to the step-size (i.e., we can assign an unchanged constant step-size for the RA-SGD algorithm during all the iterations), but the RA-SGD algorithm is disadvantageous in terms of its limited using range. In other words, the Aitken acceleration technique is only effective for those algorithms whose iterative functions are unchanged, e.g., SGD algorithm for systems without hidden variables.

Remark 8. The Aitken acceleration based methods are at least quadratically convergent. Thus, if the given algorithm is pth-order ($p\geqslant 2$) convergent, then there is no need to use the Aitken acceleration technique to improve it.

V. Examples

A. Example 1

Consider the following ARX model in [20],

$$\begin{aligned}y(t)&=-a_1y(t-1)-a_2y(t-2)+b_1u(t)+b_2u(t-1)+v(t)\\ &=-0.15y(t-1)-0.6y(t-2)+0.8u(t)+0.9u(t-1)\\ &\quad+v(t),\\ \boldsymbol{\theta}&=[a_1,a_2,b_1,b_2]^{\mathrm{T}}=[0.15,0.6,0.8,0.9]^{\mathrm{T}},\\ \boldsymbol{\varphi}(t)&=[-y(t-1),-y(t-2),u(t),u(t-1)]^{\mathrm{T}},\end{aligned}$$

where $\{u(t)\}$ is an input sequence with zero mean and unit variance, $\{v(t)\}$ is taken as a white noise sequence with zero mean and variance $\sigma^2=0.10^2$.

In simulation, let $L=500$ and the initial parameters be $\hat{\boldsymbol{\theta}}_0=\mathbf{1}/10^6$ and $\bar{\boldsymbol{\theta}}_0=\mathbf{1}/10^6$, where $\mathbf{1}=[1,1,1,1]^{\mathrm{T}}$. Apply the SGD algorithm and the corresponding RA-SGD algorithm to this ARX model (γ with different values). The estimation errors $\delta:=\|\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ (for the SGD algorithm) or $\delta:=\|\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ (for the RA-SGD algorithm) versus k are shown in Figs. 1- 4. The parameter estimates and the estimation errors are shown in Tables II-IV.

From this simulation, the following findings can be obtained.

1) When the step-size $\gamma=\frac{2.5}{\lambda_{\max}(\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L))}$, the SGD algorithm is divergent but the RA-SGD algorithm is convergent. This is verified by Fig. 1 and Table II.
2) When the step-size $\gamma=\frac{2}{\lambda_{\max}(\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L))}$, the estimates by using the SGD algorithm converge to a stationary point, while the estimates by using the RA-SGD algorithm achieve the optimal point. This is shown in Fig. 2 and Table III.
3) When the step-size is small, the convergence rate of the RA-SGD algorithm is much faster than that of the SGD algorithm. This is demonstrated in Fig. 4 and Table IV.

TABLE I
THE CONVERGENCE OF THE SGD AND RA-SGD ALGORITHMS

Algorithms	$\gamma \in (0, \frac{2}{\lambda_{\max}[\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)}$)	$\gamma = \frac{2}{\lambda_{\max}[\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]}$	$\gamma > \frac{2}{\lambda_{\max}[\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]}$
SGD	Linearly convergent	Not sure	Divergent
RA-SGD	At least quadratically convergent	At least quadratically convergent	At least quadratically convergent

TABLE II
THE PARAMETER ESTIMATES AND THEIR ESTIMATION ERRORS ($\gamma = \frac{2.5}{\lambda_{\max}[\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]}$)

	k	a_1	a_2	b_1	b_2	δ (%)
SGD	1	-0.44015	-1.70518	1.03513	0.56367	178.37448
	2	1.14086	0.44870	-0.36541	-0.82232	170.53413
	5	-2.19133	-1.79761	2.56546	2.47191	302.94712
	10	8.09778	8.13803	-6.15539	-6.74953	1112.69910
	20	111.45590	105.34076	-95.14040	-101.81418	15338.72107
	50	291721	274432	-251410	-268944	40187646
	100	145260261208	136647645409	-125184741738	-133915657335	20010603645519
RA-SGD	1	0.24943	-16.33531	0.28563	-0.40064	1255.32045
	2	0.36413	-1.28816	0.36133	-0.17152	164.38028
	5	0.01722	0.05298	0.62771	0.35273	59.37619
	10	0.00612	0.50910	0.79162	0.66190	21.62882
	20	0.10303	0.59462	0.81951	0.82810	6.51800
	50	0.14747	0.59740	0.79978	0.89760	0.32186
	100	0.14890	0.59737	0.79892	0.89967	0.22685
	True Values	0.15000	0.60000	0.80000	0.90000	

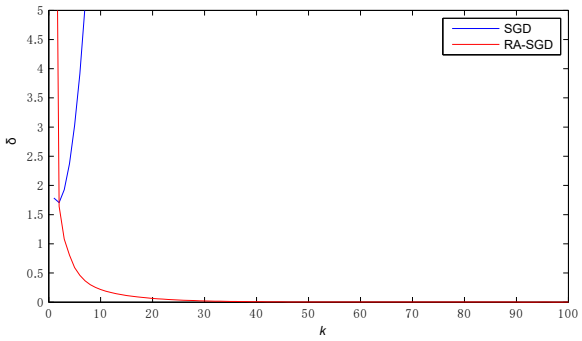


Fig. 1. The parameter estimation errors δ versus k ($\gamma = \frac{2.5}{\lambda_{\max}[\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]}$)

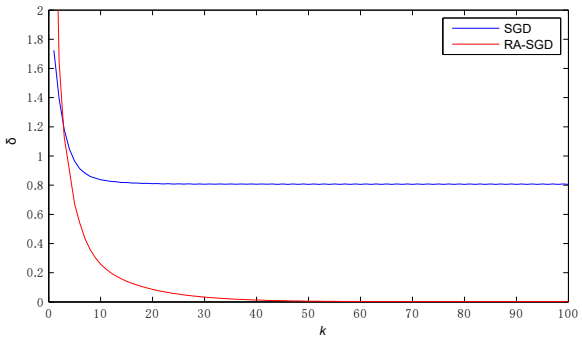


Fig. 2. The parameter estimation errors δ versus k ($\gamma = \frac{2}{\lambda_{\max}[\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]}$)

4) When the denominators in Equations (10) and (11) are extremely small, the estimation errors by using the RA-SGD algorithm will oscillate. This can be seen from Fig. 3.

TABLE III
THE PARAMETER ESTIMATES AND THEIR ESTIMATION ERRORS
($\gamma = \frac{2}{\lambda_{\max}[\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]}$)

	k	a_1	a_2	b_1	b_2	δ (%)
SGD	1	-0.22661	-1.63773	0.86576	0.35945	172.38849
	2	0.78317	-0.06774	-0.04597	-0.51097	139.24962
	5	-0.58282	-0.41866	1.15978	0.94216	96.48387
	10	0.59052	1.04980	0.28242	0.11341	83.67376
	20	0.67943	1.14332	0.31596	0.27393	80.98142
	50	0.73194	1.14857	0.29576	0.35514	80.60910
	100	0.73497	1.14852	0.29385	0.36014	80.60789
RA-SGD	1	0.32512	-3.99649	0.25289	-0.43340	356.08638
	2	0.36413	-1.28816	0.36133	-0.17152	164.38028
	5	0.04377	-0.03945	0.59838	0.29617	67.11511
	10	-0.00458	0.45950	0.77155	0.61991	25.89676
	20	0.08742	0.59025	0.82140	0.80501	8.58079
	50	0.14575	0.59743	0.80086	0.89475	0.53745
	100	0.14908	0.59737	0.79876	0.90026	0.22604
	True Values	0.15000	0.60000	0.80000	0.90000	

5) From Figs. 1-4 and Tables II-IV, we can conclude that $\gamma = \frac{2}{\lambda_{\max}(\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L))}$ is the threshold for convergence when using the SGD algorithm.

B. Example 2

Consider a three-tank system shown in Fig. 5, where the inflow q is the input u, and the second tank water level H_2 is considered as the output y. The three-tank system can be expressed by the following model [33],

$$\begin{aligned} y(t) &= -a_1 y(t-1) - a_2 y(t-2) + b_1 u(t-2) + b_2 u(t-3) + v(t) \\ &= 0.4872 y(t-1) + 0.3409 y(t-2) + 0.1088 u(t-2) + \\ &\quad 0.0476 u(t-3) + v(t), \end{aligned}$$

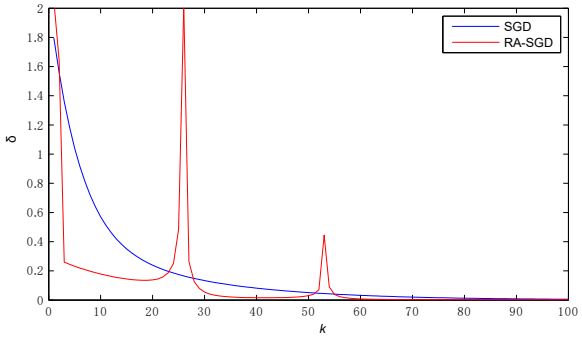


Fig. 3. The parameter estimation errors δ versus k ($\gamma = \frac{1}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$)

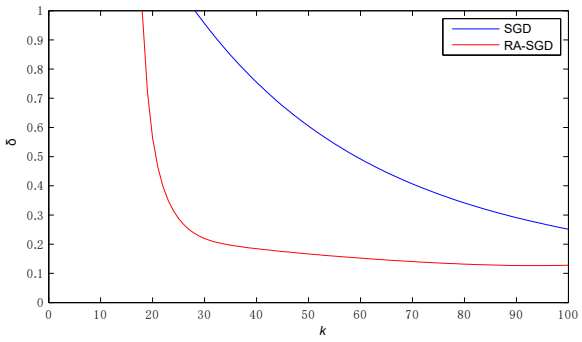


Fig. 4. The parameter estimation errors δ versus k ($\gamma = \frac{0.2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$)

TABLE IV
THE PARAMETER ESTIMATES AND THEIR ESTIMATION ERRORS
($\gamma = \frac{0.2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$)

	k	a_1	a_2	b_1	b_2	δ (%)
SGD	1	1.05459	-1.23300	-0.15048	-0.86589	211.53558
	2	0.93657	-1.25777	-0.05798	-0.75206	202.78312
	5	0.68585	-1.25025	0.13364	-0.50523	182.88640
	10	0.45931	-1.10456	0.29743	-0.26854	158.76820
	20	0.25585	-0.72532	0.43734	-0.02121	122.45418
	50	0.04495	0.02154	0.63144	0.35611	60.46093
	100	0.01246	0.45026	0.77021	0.62865	25.14419
RA-SGD	1	1.06306	-1.23712	-0.16246	-0.88255	212.98265
	2	0.36413	-1.28816	0.36133	-0.17152	164.38028
	5	0.30612	-1.26793	0.39469	-0.09146	159.48088
	10	0.21225	-1.67235	0.45421	0.05545	180.95242
	20	0.04275	1.15495	0.62185	0.41939	56.36538
	50	-0.01282	0.58683	0.84972	0.75272	16.65730
	100	-0.00454	0.59428	0.83046	0.82768	12.81055
	True Values	0.15000	0.60000	0.80000	0.90000	

$$\boldsymbol{\theta} = [a_1, a_2, b_1, b_2]^{\mathrm{T}} = [0.4872, 0.3409, 0.1088, 0.0476]^{\mathrm{T}},$$
$$\boldsymbol{\varphi}(t) = [y(t-1), y(t-2), u(t-2), u(t-3)]^{\mathrm{T}}.$$

The input $\{u(t)\}$ is a filtered random binary signal sequence and updated at every $\Delta t = 15$sec. $\{v(t)\}$ is a Gaussian white noise sequence satisfies $v(t) \sim N(0, 0.04)$. In simulation, we sample $L = 600$ input and output data. The simulation data are shown in Fig. 6. The parameter estimates and the estimation errors are shown in Figs. 7-9. From this simulation, we also can conclude that the RA-SGD algorithm is more effective than the SGD algorithm.

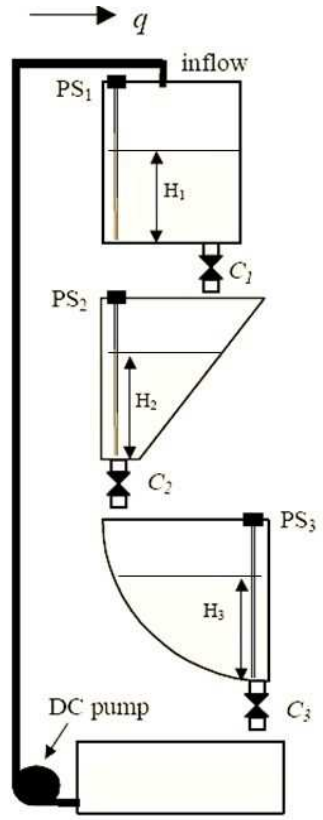


Fig. 5. The three-tank system

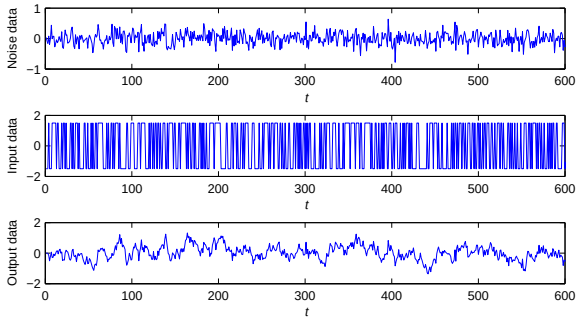


Fig. 6. The simulation data

VI. CONCLUSIONS

This paper proposes a RA-SGD algorithm for ARX models. The proposed algorithm, which has the assumption that all the elements in the information matrix are known, is based on the Aitken acceleration technique. The convergence analysis shows that the RA-SGD algorithm is at least quadratically convergent, while the SGD algorithm is only linearly convergent. When choosing a step-size for the SGD algorithm, a small step-size may lead to slow convergence rates, while a large one may cause divergence. Fortunately, the RA-SGD algorithm proposed in this paper can overcome these disadvantages. For a small step-size, the convergence rates of the SGD algorithm can be increased through the Aitken technique; while for a large one, a divergent SGD algorithm can be transformed into a convergent RA-SGD algorithm. The simulation examples also indicate that the RA-SGD algorithm has a faster convergence rate and is robust to the step-size.

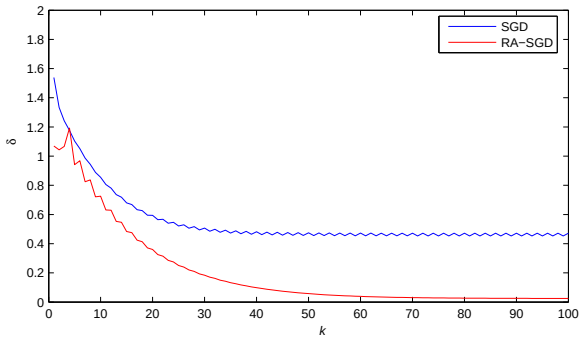


Fig. 7. The parameter estimation errors δ versus k ($\gamma = \frac{2}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$)

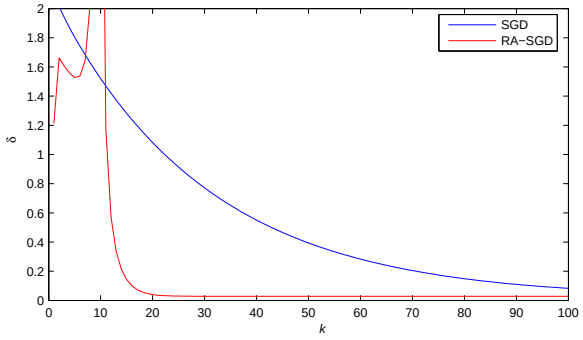


Fig. 8. The parameter estimation errors δ versus k ($\gamma = \frac{1}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$)

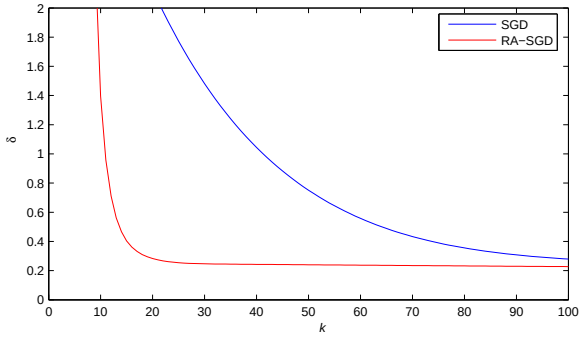


Fig. 9. The parameter estimation errors δ versus k ($\gamma = \frac{0.1}{\lambda_{\max}[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L)]}$)

Although the RA-SGD algorithm can increase the convergence rates, it also has some limitations. For example, the RA-SGD algorithm is invalid when the systems have hidden variables; the parameter estimation errors by using the Aitken based method sometimes oscillate intensively. The reasons mentioned above restrict the extensive use of this method. Thus developing some modified RA-SGD algorithms to remedy these problems is a more challenging and interesting topic in the future.

Acknowledgments

The authors would like to thank the Associate Editor and the anonymous reviewers for their constructive and helpful comments and suggestions to improve the quality of this paper.

Appendix A

Proof of Lemma 2. Assume that the iterative function is $\psi(x)$, then we have

$$x_k = \psi(x_{k-1}). \tag{20}$$

Let the fixed point be x_*, which satisfies

$$x_* = \psi(x_*).$$

Subtracting x_* on both sides of Equation (20) gets

$$x_k - x_* = \psi(x_{k-1}) - \psi(x_*) = \psi'(\varsigma)(x_{k-1} - x_*).$$

Define

$$e_k = x_k - x_*,$$
$$e_{k-1} = x_{k-1} - x_*.$$

It follows that

$$0 < \lim_{k\to\infty} \left|\frac{e_k}{e_{k-1}}\right| = \lim_{\varsigma\to x_*} |\psi'(\varsigma)| = |\psi'(x_*)| < 1.$$

Then according to Definition 1, the sequence $\{x_k\}$ is linearly convergent.

Appendix B

Proof of Lemma 4. When x_* is a fixed point of the iterative function $\phi(x)$, according to Equation (19), the following Equation holds

$$[\psi(x) - x]^2 = [x - \phi(x)][\psi(\psi(x)) - 2\psi(x) + x],$$

and it follows that

$$\lim_{x\to x_*} [x - \phi(x)] = 0,$$

which means that

$$\lim_{x\to x_*} [\psi(x) - x] = 0.$$

Thus x_* is also a fixed point of the iterative function $\psi(x)$.

When x_* is a fixed point of the iterative function $\psi(x)$, transforming Equation (19) yields

$$[x - \phi(x)] = \frac{[\psi(x) - x]^2}{\psi(\psi(x)) - 2\psi(x) + x}.$$

Taking the limit $x \to x_*$ on both sides of the above equation gives

$$\lim_{x\to x_*} [x - \phi(x)] = \lim_{x\to x_*} \frac{[\psi(x) - x]^2}{\psi(\psi(x)) - 2\psi(x) + x}. \tag{21}$$

Since the right side of the above equation is $\lim_{x\to x_*} \frac{0}{0}$. Using L' Hospital's rule for the right side of the above equation gets

$$\lim_{x\to x_*} \frac{[\psi(x) - x]^2}{\psi(\psi(x)) - 2\psi(x) + x}$$
$$= \lim_{x\to x_*} \frac{2[\psi(x) - x][\psi'(x) - 1]}{\psi'(\psi(x))\psi'(x) - 2\psi'(x) + 1}.$$

For the reason that $\lim_{x\to x_*} \psi(x) = x_*$, Equation (21) can be simplified as

$$\lim_{x\to x_*} [x - \phi(x)]$$
$$= \lim_{x\to x_*} \frac{2[\psi(x) - x][\psi'(x) - 1]}{[\psi'(x) - 1]^2}$$
$$= 0, \tag{22}$$

which means that x_* is a fixed point of the iterative function $\phi(x)$.

Appendix C

Proof of Theorem 2. According to Theorem 1, the SGD algorithm in (3) is linearly convergent, then each element in the parameter vector can be adjusted by using the Aitken technique.

Rewrite the parameter estimates $\hat{\boldsymbol{\theta}}_k$ as

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \gamma\boldsymbol{\Phi}(L)[Y(L) - \boldsymbol{\Phi}^{\mathrm{T}}(L)\hat{\boldsymbol{\theta}}_{k-1}]$$
$$= (\mathbf{I} - \gamma\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^{\mathrm{T}}(L))\hat{\boldsymbol{\theta}}_{k-1} + \gamma\boldsymbol{\Phi}(L)Y(L). \tag{23}$$

Since $\gamma\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)$ is a symmetric positive definite matrix, there exists an orthogonal matrix Q, which can keep

$$Q[\gamma\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]Q^{\mathrm{T}} = \left[\begin{array}{cccc} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_{2n} \end{array}\right], \tag{24}$$

where $\lambda_i > 0, i = 1, 2, \cdots, 2n$ are the eigenvalues of the matrix $[\gamma\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]$. Multiplying Q on both sides of Equation (23) yields

$$Q\hat{\boldsymbol{\theta}}_k = (\mathbf{I} - Q[\gamma\mathbf{\Phi}(L)\mathbf{\Phi}^{\mathrm{T}}(L)]Q^{\mathrm{T}})Q\hat{\boldsymbol{\theta}}_{k-1} + Q[\gamma\mathbf{\Phi}(L)Y(L)]. \tag{25}$$

Define

$$\begin{aligned} Q\hat{\boldsymbol{\theta}}_k &= \hat{\boldsymbol{\vartheta}}_k = [\hat{\vartheta}_k^1, \hat{\vartheta}_k^2, \cdots, \hat{\vartheta}_k^{2n}]^{\mathrm{T}} \in \mathbb{R}^{2n}, \\ Q\boldsymbol{\theta}_* &= \boldsymbol{\vartheta}_* = [\vartheta_*^1, \vartheta_*^2, \cdots, \vartheta_*^{2n}]^{\mathrm{T}} \in \mathbb{R}^{2n}, \\ Q[\gamma\mathbf{\Phi}(L)Y(L)] &= [\varrho_1, \varrho_2, \cdots, \varrho_{2n}]^{\mathrm{T}} \in \mathbb{R}^{2n}. \end{aligned}$$

Then Equation (25) is transformed into

$$\left[\begin{array}{c} \hat{\vartheta}_k^1 \\ \hat{\vartheta}_k^2 \\ \vdots \\ \hat{\vartheta}_k^{2n} \end{array}\right] = \left[\begin{array}{cccc} 1-\lambda_1 & 0 & 0 & 0 \\ 0 & 1-\lambda_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1-\lambda_{2n} \end{array}\right] \left[\begin{array}{c} \hat{\vartheta}_{k-1}^1 \\ \hat{\vartheta}_{k-1}^2 \\ \vdots \\ \hat{\vartheta}_{k-1}^{2n} \end{array}\right] + \left[\begin{array}{c} \varrho_1 \\ \varrho_2 \\ \vdots \\ \varrho_{2n} \end{array}\right]. \tag{26}$$

Each element $\hat{\vartheta}^j$ can be expressed by using an iterative function as follows

$$\hat{\vartheta}_k^j = [1-\lambda_j]\hat{\vartheta}_{k-1}^j + \varrho_j = \psi_j(\hat{\vartheta}_{k-1}^j). \tag{27}$$

According to the Aitken method, we have

$$\begin{aligned} \bar{\vartheta}_k^j = \hat{\vartheta}_k^j - \frac{(\psi_j(\hat{\vartheta}_k^j) - \hat{\vartheta}_k^j)^2}{\psi_j(\psi_j(\hat{\vartheta}_k^j)) + \hat{\vartheta}_k^j - 2\psi_j(\hat{\vartheta}_k^j)}, \\ j = 1, \cdots, 2n. \end{aligned} \tag{28}$$

Equation (28) is equivalent to the following iterative function

$$\bar{\vartheta}_k^j = \phi_j(\hat{\vartheta}_k^j),$$

where

$$\phi_j(\hat{\vartheta}_k^j) = \hat{\vartheta}_k^j - \frac{(\psi_j(\hat{\vartheta}_k^j) - \hat{\vartheta}_k^j)^2}{\psi_j(\psi_j(\hat{\vartheta}_k^j)) + \hat{\vartheta}_k^j - 2\psi_j(\hat{\vartheta}_k^j)},$$

and its first derivative is

$$[\phi(\hat{\vartheta}_k^j)]' = 1 - \frac{\alpha(\hat{\vartheta}_k^j)}{\beta(\hat{\vartheta}_k^j)}, \tag{29}$$

in which

$$\begin{aligned} \alpha(\hat{\vartheta}_k^j) &= 2[\psi(\hat{\vartheta}_k^j) - \hat{\vartheta}_k^j][\psi'(\hat{\vartheta}_k^j) - 1][\psi(\psi(\hat{\vartheta}_k^j)) - 2\psi(\hat{\vartheta}_k^j) + \hat{\vartheta}_k^j] - \\ &\quad [\psi(\hat{\vartheta}_k^j) - \hat{\vartheta}_k^j]^2[\psi'(\psi(\hat{\vartheta}_k^j))\psi'(\hat{\vartheta}_k^j) - 2\psi'(\hat{\vartheta}_k^j) + 1], \\ \beta(\hat{\vartheta}_k^j) &= [\psi(\psi(\hat{\vartheta}_k^j)) - 2\psi(\hat{\vartheta}_k^j) + \hat{\vartheta}_k^j]^2. \end{aligned}$$

Taking the limit $\hat{\vartheta}_k^j \to \vartheta_*^j$ gives

$$\lim_{\hat{\vartheta}_k^j \to \vartheta_*^j} [\phi(\hat{\vartheta}_k^j)]' = 1 - \lim_{\hat{\vartheta}_k^j \to \vartheta_*^j} \frac{0}{0}.$$

The second part on the right side of Equation (29) is $\lim_{\hat{\vartheta}_k^j \to \vartheta_*^j} \frac{0}{0}$.

For the reason that $\psi_j^{'}(\hat{\vartheta}^j) \neq 1$, then Using L' Hospital's rule for this part gets

$$\lim_{\hat{\vartheta}_k^j \to \vartheta_*^j} \frac{\alpha(\hat{\vartheta}_k^j)}{\beta(\hat{\vartheta}_k^j)} = \lim_{\hat{\vartheta}_k^j \to \vartheta_*^j} \frac{(\psi'(\hat{\vartheta}_k^j) - 1)^2}{(\psi'(\hat{\vartheta}_k^j) - 1)^2} = 1,$$

which means that

$$\lim_{\hat{\vartheta}_k^j \to \vartheta_*^j} [\phi(\hat{\vartheta}_k^j)]' = 0.$$

This shows that the iterative function is convergent. Furthermore, if the second-order derivation of the function $\phi(\hat{\vartheta}_k^j)$ is

$$\lim_{\hat{\vartheta}_k^j \to \vartheta_*^j} [\phi(\hat{\vartheta}_k^j)]^{(2)} \neq 0.$$

Then the sequence $\{\hat{\vartheta}_k^j\}$ is quadratically convergent. If the p-order ($p \geqslant 2$) derivation of the function $\phi(\hat{\vartheta}_k^j)$ is

$$\lim_{\hat{\vartheta}_k^j \to \vartheta_*^j} [\phi(\hat{\vartheta}_k^j)]^{(p)} \neq 0,$$

the sequence $\{\hat{\vartheta}_k^j\}$ is pth-order convergent.

Since $Q\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\vartheta}}_k$ and Q is an orthogonal matrix, the sequence $\{\hat{\boldsymbol{\theta}}_k\}$ has the same convergence rate as $\{\hat{\boldsymbol{\vartheta}}_k\}$. It shows that the RA-SGD algorithm is at least quadratically convergent.

REFERENCES

[1] Y. Zhao, J. Zhao, J. Fu, Y. Shi, and C. Chen, "Rate bumpless transfer control for switched linear systems with stability and its application to aero-engine control design," *IEEE Trans. Ind. Electron.*, vol. 67, no. 6, pp. 4900-4910, 2020.

[2] J. Fu, R.C. Ma, T.Y. Chai, and Z.T. Hu, "Dwell-time-based standard H_∞ control of switched systems without requiring internal stability of subsystems," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 3019-3025, 2019.

[3] T.S. Chen, B.S. Thomas, O. Henrik, and L. Ljung, "Decentralized particle filter with arbitrary state decomposition," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 465-478, 2011.

[4] W. Chen, D.R. Ding, X.H. Ge, Q.L. Han, and G.L. Wei, "H_∞ containment control of multiagent systems under event-triggered communication scheduling: the finite-horizon case," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1372-1382, 2020.

[5] Y. Zhao, A. Fatehi, and B. Huang, "Robust estimation of ARX models with time varying time delays using variational Bayesian," *IEEE Trans. Cybern.*, vol. 8, no. 2, pp. 532-542, 2018.

[6] T.S. Chen, M.S. Andersen, L. Ljung, A. Chiuso, and G. Pillonetto, "System identification via sparse multiple Kernel-based regularization using sequential convex optimization techniques," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 2933-2945, 2014.

[7] F. Ding, L. Xu, and Q.M. Zhu, "Performance analysis of the generalised projection identification for time-varying systems," *IET Control Theory and Applications*, vol. 10, no. 18, pp. 2506-2514, 2016.

[8] T. Söderström and U. Soverini, "Errors-in-variables identification using maximum likelihood estimation in the frequency domain," *Automatica*, vol. 79, pp. 131-143, 2017.

[9] F. Ding, L. Lv, J. Pan, X.K. Wan, and X.B. Jin, "Two-stage gradient-based iterative estimation methods for controlled autoregressive systems using the measurement data," *Int. J. Control, Autom. Syst.*, vol. 18, no. 4, pp. 886-896, 2020.

[10] M. Gan, C.L.P. Chen, G.Y. Chen, and L. Chen, "On some separated algorithms for separable nonlinear squares problems," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2866-2874, 2018.

[11] G.Y. Chen, M. Gan, C.L.P. Chen, and H.X. Li, "Basis function matrix-based flexible coefficient autoregressive models: A framework for time series and nonlinear system modeling," *IEEE Trans. Cybern.*, 2020. DOI: 10.1109/TCYB.2019.2900469

[12] 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, 2020.

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

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

[15] F. Ding, X. Zhang, and L. Xu, "The innovation algorithms for multivariable state-space models," *Int. J. Adapt. Control*, vol. 33, no. 11, pp. 1601-1608, 2019.

[16] F. Ding, L. Xu, D.D. Meng, X.B. Jin, A. Alsaedi, and T. Hayat, "Gradient estimation algorithms for the parameter identification of bilinear systems using the auxiliary model," *J. Comput. Appl. Math.*, vol. 369, p. 112575, 2020.

[17] S. Abbasbandy, A. Jafarian, and R. Ezzati, "Conjugate gradient method for fuzzy symmetric positive definite system of linear equations," *Appl. Math. Comput.*, vol. 171, no. 2, pp. 1184-1191, 2005.

[18] J.X. Ma, W.L. Xiong, F. Ding, A. Alsaedi, and T. Hayat, "Data filtering based forgetting factor stochastic gradient algorithm for Hammerstein systems with saturation and preload nonlinearities," *J. Frankl. Inst.*, vol. 353, no. 16, pp. 4280-4299, 2016.

[19] J. Chen and F. Ding, "Modified stochastic gradient algorithms with fast convergence rates," *J. Vib. Control*, vol. 17, no. 9, pp. 1281-1286, 2011.

[20] J. Chen, F. Ding, Y.J. Liu, and Q.M. Zhu, "Multi-step-length gradient iterative algorithm for equation-error type models," *Syst. Control Lett.*, vol. 115, pp. 15-21, 2018.

[21] M. Schmidt, N.L. Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Math. Program.*, vol. 162, no. 1-2, pp. 83-112, 2017.

[22] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *News Physiol. Sci.*, vol. 1, no. 3, pp. 315-323, 2013.

[23] I. Ramire and T. Helfer, "Iterative residual-based vector methods to accelerate fixed point iterations," *Comput. Math. Appl.*, vol. 70, no. 9, pp. 2210-2226, 2015.

[24] C. Brezinski, "Convergence acceleration during the 20th century," *J. Comput. Appl. Math.*, vol. 122, no. 1-2, pp. 1-21, 2000.

[25] I. Pavaloiu and E. Catina, "On a robust Aitken-Newton method based on the Hermite polynomial" *Appl. Math. Comput.*, vol. 287-288, pp. 224-231, 2016.

[26] O. Bumbariu, "A new Aitken type method for accelerating iterative sequences," *Appl. Math. Comput.*, vol. 219, no. 1, pp. 78-82, 2012.

[27] C. Wang and K. Li, "Aitken-based stochastic gradient algorithm for ARX models with time delay," *Circuits Syst. Signal Process.*, vol. 38, no. 6, pp. 2863-2876, 2018.

[28] F. Ding, X.M. Liu, H.B. Chen, and G.Y. Yao, "Hierarchical gradient based and hierarchical least squares based iterative parameter identification for CARARMA systems," *Signal Process.*, vol. 97, pp. 31-39, 2014.

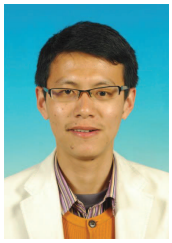
[29] F. Ding, X.H. Wang, L. Mao, and L. Xu, "Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering," *Digit. Signal Process.*, vol. 62, pp. 211-223, 2017.

[30] J. Chen, Q.M. Zhu, M.F. Hu, L.X. Guo, and P. Narayan, "Improved gradient descent algorithms for time-delay rational state-space systems: intelligent search method and momentum method," *Nonlinear Dyn.*, 2020. DOI: 10.1007/s11071-020-05755-8

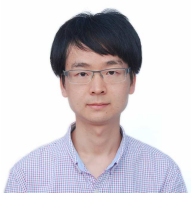
[31] X.Q. Tang, L.J. Wang, and W. Fang, "*Numerical Calculation*." Science Press, 2015.

[32] E.W. Cheney and D.R. Kincaid, "*Numerical Mathematics and Computing*." Brooks Cole, 2007.

[33] 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, 2013.

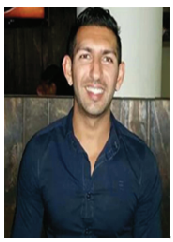


Jing Chen received his B.Sc. degree in the School of Mathematical Science and M.Sc. degree in the School of Information Engineering from Yangzhou University (Yanghzou, China) in 2003 and 2006, respectively, and received his Ph.D. degree in the School of Internet of Things Engineering, Jiangnan University (Wuxi, China) in 2013. He is currently an associate 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 dontrol and system identification.



Min Gan received the B. S. degree in Computer Science and Engineering from Hubei University of Technology, Wuhan, China, in 2004, and the Ph.D. degree in Control Science and Engineering from Central South University, Changsha, China, in 2010. He is currently a professor in the College of Computer Science & Technology, Qingdao University, Qingdao, China. His current research interests include statistical learning, system identification and nonlinear time series analysis.

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. His other research interest is in investigating electrodynamics of acupuncture points and sensory stimulation effects in human body, modelling of human meridian systems, and building up electro-acupuncture instruments. He has published over 200 papers on these topics, edited five Springer books and one book for the other publisher, 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, Member of Editorial Committee of Chinese Journal of Scientific Instrument, and Editor of Elsevier book series of Emerging Methodologies and Applications in Modelling, Identification and Control. He is the founder and president of series annual International Conference on Modelling, Identification and Control.



Pritesh Narayan is an Associate Head of Department and the Head of the Aerospace, Aviation and Manuagement Cluster at the University of the West of England (UWE) Bristol. He also a member of the Engineering Modelling and Simulation research Group (EMSG) which explores complex engineering design, analysis and control relevant issues which occur in practical engineering systems. His research interests include guidance, navigation and control of fixed wing aircraft and autonomous Unmanned Aerial Vehicle (UAV) planning and decision making.

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.