

Research Article

Parameter Identification of ARX Models Based on Modified Momentum Gradient Descent Algorithm

Quan Tu,¹ Yingjiao Rong ,² and Jing Chen ¹

¹School of Science, Jiangnan University, Wuxi 214122, China

²The Science and Technology on Near-Surface Detection Laboratory, Wuxi 214028, China

Correspondence should be addressed to Yingjiao Rong; rongyj19761229@126.com

Received 15 February 2020; Accepted 22 May 2020; Published 17 June 2020

Academic Editor: Quanmin Zhu

Copyright © 2020 Quan Tu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The parameter estimation problem of the ARX model is studied in this paper. First, some traditional identification algorithms are briefly introduced, and then a new parameter estimation algorithm—the modified momentum gradient descent algorithm—is developed. Two gradient directions with their corresponding step sizes are derived in each iteration. Compared with the traditional parameter identification algorithms, the modified momentum gradient descent algorithm has a faster convergence rate. A simulation example shows that the proposed algorithm is effective.

1. Introduction

There are many identification algorithms which can estimate the parameters of linear models and nonlinear models, such as the coupled identification algorithms [1, 2], the filtered identification algorithms [3, 4], and the hierarchical identification algorithms [5–7]. The autoregressive exogenous (ARX) model is based on the traditional autoregressive model, adding measurable external inputs at various times to generate output. Such a model is widely used in engineering practice. For example, Naveros used the ARX model to identify physical parameters of walls [8], Qin et al. applied the ARX model to control the magnetic levitation ball system [9], Haddouche et al. utilized the ARX model to control the gas condition tower [10]. Since a robust controller often has the assumption that the structures of the systems should be given in prior [11–13], the system identification plays an important role in control engineering. Its basic idea is to use the identification algorithms to determine a mathematical model [14–17] and by which the behavior of the systems can be predicted.

The gradient descent algorithm is usually used for ARX model identification. It can effectively reduce the computational efforts but with slow convergence rates [18, 19]. The gradient descent algorithm includes two steps: the first is to determine a direction, which is called negative gradient, and

the second is to determine a suitable step size for the direction. Besides, the least squares algorithm is another widely used method in system identification, which has a faster convergence rate [20–26]. However, the least squares algorithm has heavy computational efforts and needs to solve a derivative function. Therefore, it is inefficient for some models with complex nonlinear structures.

In order to determine the step size of the gradient descent algorithm, the root of a higher-order equation needs to be calculated, which is challenging/impossible. Fortunately, the stochastic gradient (SG) algorithms [27, 28] avoid the root calculation by updating the parameters in each sampling time with only one set input-output data. It can be widely used in engineering practice for its simple structure. However, only one set of data is used at each sampling instant; the convergence rate of the SG algorithm is slow. To improve the convergence rate, Ding et al. first proposed a multi-innovation stochastic gradient algorithm and a multi-innovation least squares algorithm for linear regression models [29, 30], which have quick convergence rates. The conjugate gradient descent method is another method which has quicker convergence rate when comparing with the gradient descent algorithm, but it is only available for offline identification [31–34]. Inspired by the conjugate gradient descent algorithm, the focus of this paper is to propose a

modified momentum gradient descent algorithm, which has a quicker convergence rate and no root calculation.

The remainder of this paper is organized as follows. Section 2 introduces the ARX model and the traditional SG algorithm. The multi-innovation stochastic gradient algorithm is presented in Section 3. In Section 4, a modified momentum gradient descent algorithm is developed. A simulation example is given in Section 5. Finally, the conclusions and future directions are summarized in Section 6.

2. Stochastic Gradient Descent Algorithm

Consider the following ARX model:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \dots + a_n y(t-n) + b_1 u(t-1) + b_2 u(t-2) + \dots + b_m u(t-m) + v(t). \quad (4)$$

Let θ^* be the true value, $\hat{\theta}$ be the estimated one:

$$\begin{aligned} \theta^* &= [a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m]^T \in \mathbb{R}^{m+n}, \\ \hat{\theta} &= [\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n, \hat{b}_1, \hat{b}_2, \dots, \hat{b}_m]^T \in \mathbb{R}^{m+n}, \end{aligned} \quad (5)$$

and $\varphi(t)$ is the information vector:

$$\begin{aligned} \varphi(t) &= [y(t-1), y(t-2), \dots, y(t-n), \\ &u(t-1), u(t-2), \dots, u(t-m)]^T \in \mathbb{R}^{m+n}. \end{aligned} \quad (6)$$

Define the cost function as follows:

$$J(\hat{\theta}) = \frac{1}{2} (y(t) - \varphi^T(t) \hat{\theta})^2. \quad (7)$$

To obtain the minimum value of $J(\hat{\theta})$, let the iteration function be

$$F(\hat{\theta}) = \hat{\theta} + a\mathbf{p}. \quad (8)$$

When the estimated parameter vector $\hat{\theta}$ converges to the true value θ^* ,

$$\hat{\theta} = F(\hat{\theta}) = \hat{\theta} + a\mathbf{p}, \quad (9)$$

$$\mathbf{p} = -\Delta J = \varphi(t) (y(t) - \varphi^T(t) \hat{\theta}), \quad (10)$$

where \mathbf{p} is the negative gradient direction of J and a is the step size. Substituting equation (6) into equation (4) yields

$$J(a) = \frac{1}{2} (y(t) - \varphi^T(t) (\hat{\theta} + a\mathbf{p}))^2. \quad (11)$$

In order to get the minimum value of $J(a)$, use

$$\frac{\partial J}{\partial a} = (y(t) - \varphi^T(t) (\hat{\theta} + a\mathbf{p})) (-\varphi^T(t) \mathbf{p}), \quad (12)$$

and let $(\partial J / \partial a) = 0$. The steepest descent algorithm can be obtained:

$$A(z)y(t) = B(z)u(t) + v(t), \quad (1)$$

where $y(t)$ is the output, $u(t)$ is the input, $v(t)$ is the noise, and $A(z)$ and $B(z)$ are polynomials:

$$A(z) = 1 + \bar{a}_1 z^{-1} + \bar{a}_2 z^{-2} + \dots + \bar{a}_n z^{-n}, \quad (2)$$

$$B(z) = b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}. \quad (3)$$

Take equations (2) and (3) into equation (1), and let $a_i = -\bar{a}_i$, $i = 1, 2, \dots, n$, the ARX model can be written by

$$\hat{\theta}_t = F(\hat{\theta}_{t-1}) = \hat{\theta}_{t-1} + \frac{\varphi(t)(y(t) - \varphi^T(t) \hat{\theta}_{t-1})}{\varphi^T(t) \varphi(t)}. \quad (13)$$

Remark 1. When $\hat{\theta}_t$ is close to the true value, the calculated step size would be imprecise, which will cause the error to fluctuate. Therefore, the steepest descent algorithm is inefficient.

The SG algorithm proposed in the following can deal with this problem:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \frac{\varphi(t)}{a't_t} (y(t) - \varphi^T(t) \hat{\theta}_{t-1}),$$

$$\begin{aligned} \varphi(t) &= [y(t-1), y(t-2), \dots, y(t-n), \\ &u(t-1), u(t-2), \dots, u(t-m)]^T, \end{aligned} \quad (14)$$

$$a'_t = a'_{t-1} + \varphi^T(t) \varphi(t),$$

$$a'_0 = 0.$$

Remark 2. The step size will be reduced with the increase in time. When $\hat{\theta}_t$ is close to the true value, the smaller step size reduces the fluctuation dramatically.

3. Two-Innovation Stochastic Gradient (TI-SG) Descent Algorithm

Because of the slow convergence rate of the SG algorithm, Ding proposed a multi-innovation stochastic gradient (MI-SG) algorithm in [6]. As a special case of the MI-SG algorithm, when two sets of input-output data are performed in each iteration, we term it as two-innovation stochastic gradient (TI-SG) algorithm.

For the ARX model, two sets input-output data are collected in each iteration as follows:

$$\begin{aligned}\boldsymbol{\varphi}(t-1) &= [y(t-2), y(t-3), \dots, y(t-n-1), u(t-2), u(t-3), \dots, u(t-m-1)]^T, \\ \boldsymbol{\varphi}(t) &= [y(t-1), y(t-2), \dots, y(t-n), u(t-1), u(t-2), \dots, u(t-m)]^T.\end{aligned}\quad (15)$$

Establishing the following two functions $J_1(\hat{\boldsymbol{\theta}})$ and $J_2(\hat{\boldsymbol{\theta}})$, we get

$$\begin{aligned}J_1(\hat{\boldsymbol{\theta}}) &= \frac{1}{2}(y(t-1) - \boldsymbol{\varphi}^T(t-1)\hat{\boldsymbol{\theta}})^2, \\ J_2(\hat{\boldsymbol{\theta}}) &= \frac{1}{2}(y(t) - \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}})^2.\end{aligned}\quad (16)$$

We can calculate the negative gradient directions \mathbf{p}_1 and \mathbf{p}_2 , respectively:

$$\begin{aligned}\mathbf{p}_1 &= -\Delta J_1 = \boldsymbol{\varphi}(t-1)(y(t-1) - \boldsymbol{\varphi}^T(t-1)\hat{\boldsymbol{\theta}}), \\ \mathbf{p}_2 &= -\Delta J_2 = \boldsymbol{\varphi}(t)(y(t) - \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}}).\end{aligned}\quad (17)$$

The cost function is established as follows:

$$J(\hat{\boldsymbol{\theta}}) = J_1(\hat{\boldsymbol{\theta}}) + J_2(\hat{\boldsymbol{\theta}}) = \frac{1}{2}(y(t-1) - \boldsymbol{\varphi}^T(t-1)\hat{\boldsymbol{\theta}})^2 + \frac{1}{2}(y(t) - \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}})^2.\quad (18)$$

Let the iteration function be

$$F(\hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\theta}} + a(\mathbf{p}_1 + \mathbf{p}_2).\quad (19)$$

Update the parameters $\hat{\boldsymbol{\theta}} = F(\hat{\boldsymbol{\theta}})$, then the cost function is

$$J(a) = \sum_{i=0}^1 \frac{1}{2}(y(t-i) - \boldsymbol{\varphi}^T(t-i)(\hat{\boldsymbol{\theta}} + a(\mathbf{p}_1 + \mathbf{p}_2)))^2.\quad (20)$$

$$\frac{\partial J}{\partial a} = \sum_{i=0}^1 (y(t-i) - \boldsymbol{\varphi}^T(t-i)(\hat{\boldsymbol{\theta}}_{t-1} + a(\mathbf{p}_1 + \mathbf{p}_2))) \cdot (-\boldsymbol{\varphi}^T(t-i)(\mathbf{p}_1 + \mathbf{p}_2)).\quad (22)$$

Let $(\partial J/\partial a)$ equal 0, then

$$a = \frac{(\mathbf{p}_1 + \mathbf{p}_2)^T (\mathbf{p}_1 + \mathbf{p}_2)}{\sum_{i=0}^1 (\boldsymbol{\varphi}^T(t-i)(\mathbf{p}_1 + \mathbf{p}_2))^2}.\quad (23)$$

There are two ways to calculate the step size a :

- (1) The two-innovation stochastic gradient descent algorithm has the same step size as that in the SG algorithm. Let the initial value of the step size be 0. The TI-SG algorithm can be designed as

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \frac{\mathbf{p}_1 + \mathbf{p}_2}{a'_t},$$

$$\mathbf{p}_1 = \boldsymbol{\varphi}(t-1)(y(t-1) - \boldsymbol{\varphi}^T(t-1)\hat{\boldsymbol{\theta}}_{t-1}),$$

$$\mathbf{p}_2 = \boldsymbol{\varphi}(t)(y(t) - \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}}_{t-1}),$$

$$\begin{aligned}\boldsymbol{\varphi}(t-1) &= [y(t-2), y(t-3), \dots, y(t-n-1), \\ &u(t-2), u(t-3), \dots, u(t-m-1)]^T,\end{aligned}\quad (21)$$

$$\begin{aligned}\boldsymbol{\varphi}(t) &= [y(t-1), y(t-2), \dots, y(t-n), u(t-1), \\ &u(t-2), \dots, u(t-m)]^T,\end{aligned}$$

$$a'_t = a'_{t-1} + \boldsymbol{\varphi}^T(t)\boldsymbol{\varphi}(t),$$

$$a'_0 = 0.$$

- (2) The other method is to calculate the optimal step size, which is called modified two-innovation stochastic gradient (MT-SG) descent algorithm.

The MT-SG algorithm can be designed as

$$\begin{aligned}\hat{\boldsymbol{\theta}}_t &= \hat{\boldsymbol{\theta}}_{t-1} + a_t(\mathbf{p}_1 + \mathbf{p}_2), \\ \mathbf{p}_1 &= \boldsymbol{\varphi}(t-1)(y(t-1) - \boldsymbol{\varphi}^T(t-1)\hat{\boldsymbol{\theta}}_{t-1}), \\ \mathbf{p}_2 &= \boldsymbol{\varphi}(t)(y(t) - \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}}_{t-1}), \\ \boldsymbol{\varphi}(t-1) &= [y(t-2), y(t-3), \dots, y(t-n-1), u(t-2), u(t-3), \dots, u(t-m-1)]^T, \\ \boldsymbol{\varphi}(t) &= [y(t-1), y(t-2), \dots, y(t-n), u(t-1), u(t-2), \dots, u(t-m)]^T, \\ a_t &= \frac{(\mathbf{p}_1 + \mathbf{p}_2)^T (\mathbf{p}_1 + \mathbf{p}_2)}{\sum_{i=0}^1 (\boldsymbol{\varphi}^T(t-i)(\mathbf{p}_1 + \mathbf{p}_2))^2}.\end{aligned}\quad (24)$$

Remark 3. The traditional two-innovation algorithm and the modified two-innovation algorithm use two gradients and assume that the two gradient directions have the same step size. Although the computational effort is reduced, it is not optimal. Because each gradient direction plays a different role in estimating the parameters, it is necessary to consider assigning different weights to each gradient.

Remark 4. Compared with the traditional two-innovation method, the modified two-innovation method calculates the optimal step size in each sampling instant. Therefore, the

modified two-innovation algorithm has a faster convergence rate but with heavier computational efforts.

4. Modified Momentum Gradient Descent Algorithm (MMG)

Before introducing the modified momentum gradient descent algorithm, we first introduce the conjugate gradient descent algorithm.

Assume that we have collected L input-output data. The collected information vectors and outputs are $\Phi(L)$ and $\mathbf{Y}(L)$, respectively,

$$\begin{aligned}\boldsymbol{\varphi}(t) &= [y(t-1), y(t-2), \dots, y(t-n), u(t-1), u(t-2), \dots, u(t-m)]^T \in \mathbb{R}^{m+n}, \\ \boldsymbol{\Phi}(L) &= [\boldsymbol{\varphi}^T(t+1), \boldsymbol{\varphi}^T(t+2), \dots, \boldsymbol{\varphi}^T(t+L)]^T \in \mathbb{R}^{L \times (m+n)}, \\ \mathbf{Y}(L) &= [y(t+1), y(t+2), \dots, y(t+L)]^T \in \mathbb{R}^L.\end{aligned}\quad (25)$$

Set up the cost function as follows:

$$J(\hat{\boldsymbol{\theta}}) = \frac{1}{2}(\mathbf{Y}(t) - \boldsymbol{\Phi}(L)\hat{\boldsymbol{\theta}})^2. \quad (26)$$

To calculate the minimum value of $J(\hat{\boldsymbol{\theta}})$, simply make $(\partial J / \partial \hat{\boldsymbol{\theta}}) = 0$:

$$\frac{\partial J}{\partial \hat{\boldsymbol{\theta}}} = -\boldsymbol{\Phi}^T(L)(\mathbf{Y}(t) - \boldsymbol{\Phi}(L)\hat{\boldsymbol{\theta}}). \quad (27)$$

Let $\mathbf{A} = \boldsymbol{\Phi}^T(L)\boldsymbol{\Phi}(L)$ and $\mathbf{b} = \boldsymbol{\Phi}^T(L)\mathbf{Y}(t)$. When the order of $\boldsymbol{\Phi}(L)$ is greater than $n+m$, it is easy to know that \mathbf{A} is a symmetric positive definite matrix, and $\mathbf{A} \in \mathbb{R}^{(m+n) \times (m+n)}$ and $\mathbf{b} \in \mathbb{R}^{m+n}$.

Using the conjugate gradient descent method to solve higher-order matrix equations, let $\mathbf{r} = \mathbf{b} - \mathbf{A}\hat{\boldsymbol{\theta}}$, where \mathbf{r} is the current negative gradient direction. Reconcile the previous iteration direction \mathbf{p}' with the current negative gradient direction \mathbf{r} as the new iteration direction \mathbf{p} , which is $\mathbf{p} = \mathbf{r} + \beta\mathbf{p}'$. Making \mathbf{p}' and \mathbf{p} conjugate about \mathbf{A} , that is $\mathbf{p}'^T\mathbf{A}\mathbf{p}' = 0$, we have

$$\beta = -\frac{\mathbf{r}^T\mathbf{A}\mathbf{p}'}{\mathbf{p}'^T\mathbf{A}\mathbf{p}'}. \quad (28)$$

Let the iteration function be

$$F(\hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\theta}} + a\mathbf{p}, \quad (29)$$

where a is the step size and \mathbf{p} is the iteration direction, then

$$J(a) = \frac{1}{2}(\mathbf{Y}(t) - \boldsymbol{\Phi}(L)(\hat{\boldsymbol{\theta}} + a\mathbf{p}))^2. \quad (30)$$

Calculating the minimum value of $J(a)$ and letting

$$\frac{\partial J}{\partial a} = (-\boldsymbol{\Phi}(L)\mathbf{p})^T(\mathbf{Y}(t) - \boldsymbol{\Phi}(L)(\hat{\boldsymbol{\theta}} + a\mathbf{p})) = 0, \quad (31)$$

yield

$$a = \frac{\mathbf{p}^T\boldsymbol{\Phi}^T(L)(\mathbf{Y}(t) - \boldsymbol{\Phi}(L)\hat{\boldsymbol{\theta}})}{\mathbf{p}^T\boldsymbol{\Phi}^T(L)\boldsymbol{\Phi}(L)\mathbf{p}} = \frac{\mathbf{p}^T(\mathbf{b} - \mathbf{A}\hat{\boldsymbol{\theta}})}{\mathbf{p}^T\mathbf{A}\mathbf{p}} = \frac{\mathbf{p}^T\mathbf{r}}{\mathbf{p}^T\mathbf{A}\mathbf{p}}. \quad (32)$$

Let $\hat{\boldsymbol{\theta}}$ satisfy

$$\hat{\boldsymbol{\theta}}_k = F(\hat{\boldsymbol{\theta}}_{k-1}) = \hat{\boldsymbol{\theta}}_{k-1} + \frac{\mathbf{p}_k^T\mathbf{r}_k}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k}\mathbf{p}_k. \quad (33)$$

The conjugate gradient descent algorithm can be designed as

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + a\mathbf{p}_k,$$

$$\mathbf{p}_k = \mathbf{r}_k - \frac{\mathbf{r}_k^T\mathbf{A}\mathbf{p}_{k-1}}{\mathbf{p}_{k-1}^T\mathbf{A}\mathbf{p}_{k-1}}\mathbf{p}_{k-1},$$

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\hat{\boldsymbol{\theta}}_{k-1},$$

$$a = \frac{\mathbf{p}_k^T\mathbf{r}_k}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k}, \quad (34)$$

$$\begin{aligned}\boldsymbol{\varphi}(t) &= [y(t-1), y(t-2), \dots, y(t-n), \\ &\quad u(t-1), u(t-2), \dots, u(t-m)]^T, \boldsymbol{\Phi}(L) \\ &= [\boldsymbol{\varphi}^T(t+1), \boldsymbol{\varphi}^T(t+2), \dots, \boldsymbol{\varphi}^T(t+L)]^T, \mathbf{Y}(L) \\ &= [y(t+1), y(t+2), \dots, y(t+L)]^T, \mathbf{A} \\ &= \boldsymbol{\Phi}^T(L)\boldsymbol{\Phi}(L), \mathbf{b} = \boldsymbol{\Phi}^T(L)\mathbf{Y}(t).\end{aligned}$$

Remark 5. Here \mathbf{r} is the negative gradient direction of the current position and \mathbf{p}' is the direction of the last iteration. The current iteration direction \mathbf{p} is obtained based on \mathbf{r} and \mathbf{p}' . Compared with the traditional gradient descent method,

this method has a faster convergence rate but with heavier computational efforts.

Inspired by the conjugate gradient descent method, the modified momentum gradient descent algorithm is proposed. Its basic idea is to use two gradient directions in each iteration/sampling instant and then to assign different step sizes for each direction.

$$\begin{aligned}\boldsymbol{\varphi}(2t) &= [y(2t-1), y(2t-2), \dots, y(2t-n), u(2t-1), u(2t-2), \dots, u(2t-m)]^T, \\ \boldsymbol{\varphi}(2t+1) &= [y(2t), y(2t-1), \dots, y(2t-n+1), u(2t), u(2t-1), \dots, u(2t-m+1)]^T.\end{aligned}\quad (35)$$

Establish two cost functions $J_1(\hat{\boldsymbol{\theta}})$ and $J_2(\hat{\boldsymbol{\theta}})$ as follows:

$$J_1(\hat{\boldsymbol{\theta}}) = \frac{1}{2} (y(2t) - \boldsymbol{\varphi}^T(2t)\hat{\boldsymbol{\theta}})^2, \quad (36)$$

$$J_2(\hat{\boldsymbol{\theta}}) = \frac{1}{2} (y(2t+1) - \boldsymbol{\varphi}^T(2t+1)\hat{\boldsymbol{\theta}})^2.$$

Using $J_1(\hat{\boldsymbol{\theta}}), J_2(\hat{\boldsymbol{\theta}})$ to calculate the negative gradient directions $\mathbf{p}_1, \mathbf{p}_2$ yields

$$\begin{aligned}\mathbf{p}_1 &= -\Delta J_1 = \boldsymbol{\varphi}(2t)(y(2t) - \boldsymbol{\varphi}^T(2t)\hat{\boldsymbol{\theta}}), \\ \mathbf{p}_2 &= -\Delta J_2 = \boldsymbol{\varphi}(2t+1)(y(2t+1) - \boldsymbol{\varphi}^T(2t+1)\hat{\boldsymbol{\theta}}).\end{aligned}\quad (37)$$

Let the iteration function be

$$F(\hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\theta}} + a_1\mathbf{p}_1 + a_2\mathbf{p}_2. \quad (38)$$

Then, the cost functions are

$$J_1(a_1, a_2) = \frac{1}{2} (y(2t) - \boldsymbol{\varphi}^T(2t)(\hat{\boldsymbol{\theta}} + a_1\mathbf{p}_1 + a_2\mathbf{p}_2))^2,$$

$$J_2(a_1, a_2) = \frac{1}{2} (y(2t+1) - \boldsymbol{\varphi}^T(2t+1)(\hat{\boldsymbol{\theta}} + a_1\mathbf{p}_1 + a_2\mathbf{p}_2))^2. \quad (39)$$

Let $(\partial J_1/\partial a_1), (\partial J_1/\partial a_2), (\partial J_2/\partial a_1)$, and $(\partial J_2/\partial a_2)$ all be equal to 0, then

$$\begin{cases} (y(2t) - \boldsymbol{\varphi}^T(2t)(\hat{\boldsymbol{\theta}} + a_1\mathbf{p}_1 + a_2\mathbf{p}_2))(-\boldsymbol{\varphi}^T(2t)\mathbf{p}_1) = 0, \\ (y(2t) - \boldsymbol{\varphi}^T(2t)(\hat{\boldsymbol{\theta}} + a_1\mathbf{p}_1 + a_2\mathbf{p}_2))(-\boldsymbol{\varphi}^T(2t)\mathbf{p}_2) = 0, \\ (y(2t+1) - \boldsymbol{\varphi}^T(2t+1)(\hat{\boldsymbol{\theta}} + a_1\mathbf{p}_1 + a_2\mathbf{p}_2))(-\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_1) = 0, \\ (y(2t+1) - \boldsymbol{\varphi}^T(2t+1)(\hat{\boldsymbol{\theta}} + a_1\mathbf{p}_1 + a_2\mathbf{p}_2))(-\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_2) = 0.\end{cases}\quad (40)$$

Let $r_1 = y(2t) - \boldsymbol{\varphi}^T(2t)\hat{\boldsymbol{\theta}}$ and $r_2 = y(2t+1) - \boldsymbol{\varphi}^T(2t+1)\hat{\boldsymbol{\theta}}$, we have

$$\begin{aligned}a_1 &= \frac{r_1\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_2 - r_2\boldsymbol{\varphi}^T(2t)\mathbf{p}_2}{\boldsymbol{\varphi}^T(2t)\mathbf{p}_1\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_2 - \boldsymbol{\varphi}^T(2t+1)\mathbf{p}_1\boldsymbol{\varphi}^T(2t)\mathbf{p}_2}, \\ a_2 &= \frac{r_2\boldsymbol{\varphi}^T(2t)\mathbf{p}_1 - r_1\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_1}{\boldsymbol{\varphi}^T(2t)\mathbf{p}_1\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_2 - \boldsymbol{\varphi}^T(2t+1)\mathbf{p}_1\boldsymbol{\varphi}^T(2t)\mathbf{p}_2}.\end{aligned}\quad (41)$$

The MMG algorithm is listed as follows:

When using the TI-SG algorithm method, a set of repeated data during the neighbouring two sampling instants will be involved, which causes the step size unsolvable. To overcome this difficulty, a new method is developed. For the ARX model, collect two sets of information vectors and two outputs in each iteration as $\boldsymbol{\varphi}(2t), y(2t), \boldsymbol{\varphi}(2t+1)$, and $y(2t+1)$:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_t &= \hat{\boldsymbol{\theta}}_{t-1} + a_1\mathbf{p}_1 + a_2\mathbf{p}_2, \\ \mathbf{p}_1 &= -\Delta J_1 = \boldsymbol{\varphi}(2t)r_1, \\ \mathbf{p}_2 &= -\Delta J_2 = \boldsymbol{\varphi}(2t+1)r_2, \\ r_1 &= y(2t) - \boldsymbol{\varphi}^T(2t)\hat{\boldsymbol{\theta}}_{t-1}, \\ r_2 &= y(2t+1) - \boldsymbol{\varphi}^T(2t+1)\hat{\boldsymbol{\theta}}_{t-1}, \\ a_1 &= \frac{r_1\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_2 - r_2\boldsymbol{\varphi}^T(2t)\mathbf{p}_2}{\boldsymbol{\varphi}^T(2t)\mathbf{p}_1\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_2 - \boldsymbol{\varphi}^T(2t+1)\mathbf{p}_1\boldsymbol{\varphi}^T(2t)\mathbf{p}_2}, \\ a_2 &= \frac{r_2\boldsymbol{\varphi}^T(2t)\mathbf{p}_1 - r_1\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_1}{\boldsymbol{\varphi}^T(2t)\mathbf{p}_1\boldsymbol{\varphi}^T(2t+1)\mathbf{p}_2 - \boldsymbol{\varphi}^T(2t+1)\mathbf{p}_1\boldsymbol{\varphi}^T(2t)\mathbf{p}_2}.\end{aligned}\quad (42)$$

The MMG algorithm constitutes the following steps: (Algorithm 1)

Remark 6. In each iteration, the MMG algorithm uses two directions and assigns the optimal step size for each direction. Therefore, it has a quicker convergence rate. However, some iterative algorithms [35–38] and recursive algorithms [39–42] can be extended to study the parameter identification of the ARX models in this paper.

5. Example

Consider the following ARX model:

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

$$\boldsymbol{\varphi}(t) = [y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)]^T,$$

$$\boldsymbol{\theta} = [0.3, -0.7, 0.5, 0.25, 0.17]^T.$$

(43)

The input data $u(t)$ is a random sequence with a uniform distribution on $[-0.5, 0.5]$, and $v(t)$ is Gaussian white noise with $N(0, 0.01^2)$. The simulation data are shown in Figure 1.

The SG, TI-SG, MT-SG, and MMG algorithms are used to identify the parameters of the ARX model. The parameter estimates and their estimation errors are shown in Figure 2 and Tables 1–4.

```

Initialize  $\theta_0$ 
repeat
  for  $t=0, 1, \dots, n$  do
    get  $\varphi(2t), y(2t), \varphi(2t+1)$  and  $y(2t+1)$ 
     $r_1 = (y(2t) - \varphi^T(2t)\theta_t)$ 
     $r_2 = (y(2t+1) - \varphi^T(2t+1)\theta_t)$ 
    if  $|r_1| < \text{tol}$  and  $|r_2| < \text{tol}$  then
      return  $\theta_t$ 
    else
       $\mathbf{p}_1 = r_1\varphi(2t)$ 
       $\mathbf{p}_2 = r_2\varphi(2t+1)$ 
       $D = \varphi^T(2t)\mathbf{p}_1\varphi^T(2t+1)\mathbf{p}_2 - \varphi^T(2t+1)\mathbf{p}_1\varphi^T(2t)\mathbf{p}_2$ 
       $a_1 = r_1\varphi^T(2t+1)\mathbf{p}_2 - r_2\varphi^T(2t)\mathbf{p}_2$ 
       $a_2 = r_2\varphi^T(2t)\mathbf{p}_1 - r_1\varphi^T(2t+1)\mathbf{p}_1$ 
       $\theta_{t+1} = \theta_t + (1/D)(a_1\mathbf{p}_1 + a_2\mathbf{p}_2)$ 
    end
  until convergence

```

ALGORITHM 1: Modified momentum gradient descent algorithm.

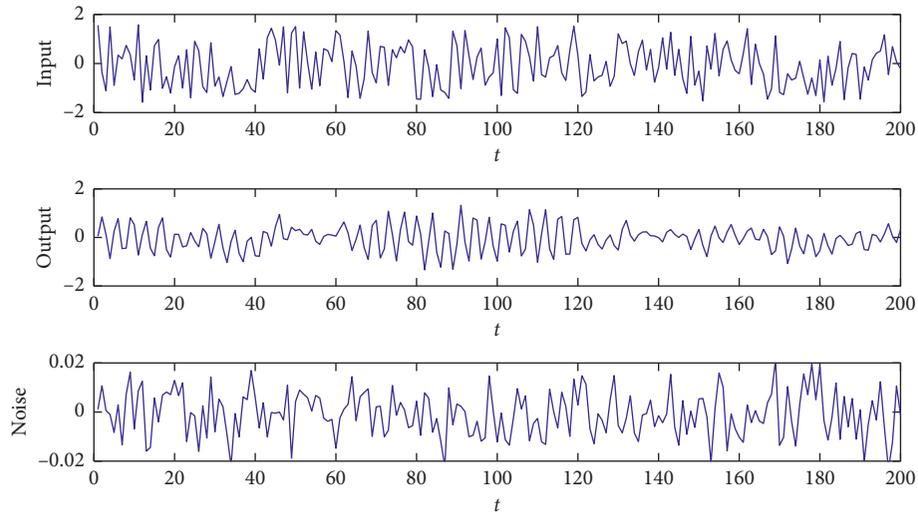


FIGURE 1: The simulation data.

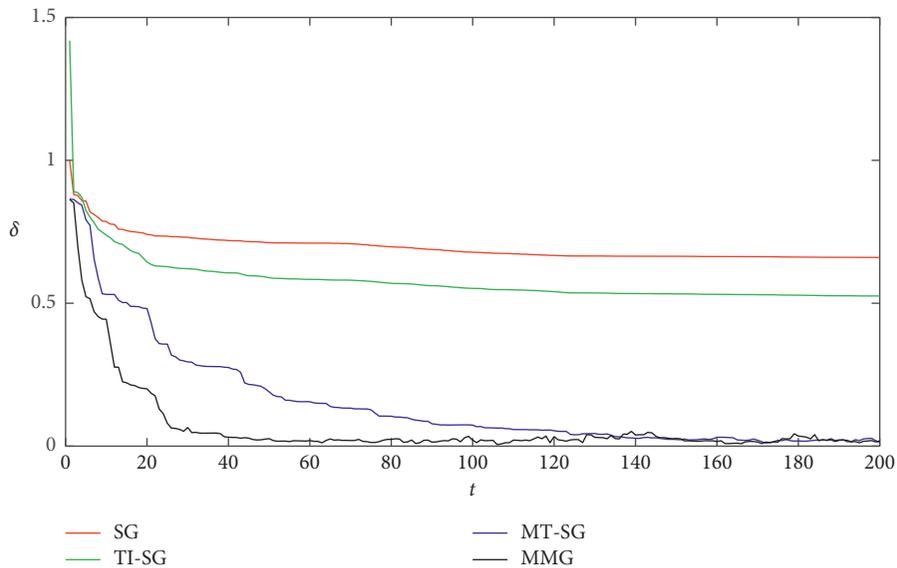


FIGURE 2: The parameter estimation errors.

TABLE 1: The parameter estimates and the estimation errors (SG).

t	θ_1	θ_2	θ_3	θ_4	θ_5	$\delta(\%)$
5	0.0090	-0.1416	-0.0195	0.3406	0.1128	85.7714
10	0.0112	-0.2101	0.0087	0.3109	0.1947	78.5886
50	-0.0030	-0.2880	0.0523	0.2917	0.2298	71.2120
100	-0.0029	-0.3280	0.0698	0.2794	0.2583	67.8208
150	-0.0065	-0.3443	0.0789	0.2750	0.2601	66.4253
200	-0.0079	-0.3484	0.0822	0.2759	0.2587	66.0179
True value	0.3000	-0.7000	0.5000	0.2500	0.1700	

TABLE 2: The parameter estimates and the estimation errors (TI-SG).

t	θ_1	θ_2	θ_3	θ_4	θ_5	$\delta(\%)$
5	0.0900	-0.2645	-0.0691	0.4479	0.3430	82.4797
10	0.0423	-0.3693	0.0104	0.4360	0.3971	73.7884
50	-0.0196	-0.4921	0.1103	0.3170	0.3050	58.9297
100	-0.0128	-0.5429	0.1327	0.2792	0.3194	55.1883
150	-0.0206	-0.5623	0.1498	0.2671	0.3000	53.2816
200	-0.0221	-0.5644	0.1580	0.2718	0.2908	52.5223
True value	0.3000	-0.7000	0.5000	0.2500	0.1700	

TABLE 3: The parameter estimates and the estimation errors (MT-SG).

t	θ_1	θ_2	θ_3	θ_4	θ_5	$\delta(\%)$
5	0.0870	-0.3185	-0.0678	0.3805	0.3885	79.2079
10	-0.0537	-0.6758	0.1948	0.2965	0.3669	53.0981
50	0.1609	-0.6486	0.3970	0.2697	0.1525	19.0093
100	0.2578	-0.7259	0.4600	0.2433	0.1997	7.3521
150	0.2925	-0.7006	0.4797	0.2456	0.1729	2.3237
200	0.3051	-0.6985	0.5004	0.2389	0.1680	1.3000
True value	0.3000	-0.7000	0.5000	0.2500	0.1700	

TABLE 4: The parameter estimates and the estimation errors (MMG).

t	θ_1	θ_2	θ_3	θ_4	θ_5	$\delta(\%)$
5	-0.0585	-0.7463	0.1831	0.3312	0.2895	52.2903
10	0.0286	-0.7357	0.1746	0.2283	0.1489	44.4168
100	0.2898	-0.6838	0.4946	0.2528	0.1684	2.0996
150	0.3037	-0.7114	0.5219	0.2534	0.1644	2.6841
200	0.3076	-0.6978	0.5114	0.2499	0.1808	1.8318
True value	0.3000	-0.7000	0.5000	0.2500	0.1700	

The relative errors of each element in the parameter vector by using these four algorithms are shown in Figure 3 ($t = 20, 50,$ and 100).

Select 100 new data based on the true model, and use the estimated models by the SG, TI-SG, MT-SG and MMG algorithms to generate the predicted outputs, respectively. The errors between the true outputs and the predicted outputs are shown in Figure 4.

Finally, a Monte Carlo experiment is performed by using the MMG algorithm (100 sets noises), and the results are shown in Figure 5.

The following conclusions can be obtained:

- (1) It can be seen from Figures 2 and 3 and Tables 2 and 3 that the MT-SG algorithm has a significantly faster speed than the original TI-SG algorithm
- (2) From Figures 2 and 3 and Tables 1–4, we can see that the MMG algorithm has the fastest convergence rates among the four algorithms
- (3) Figure 4 demonstrates that the estimated model by using the MMG algorithm is the most accurate one among these four estimated models
- (4) Figure 5 shows that the MMG algorithm is robust to the noises

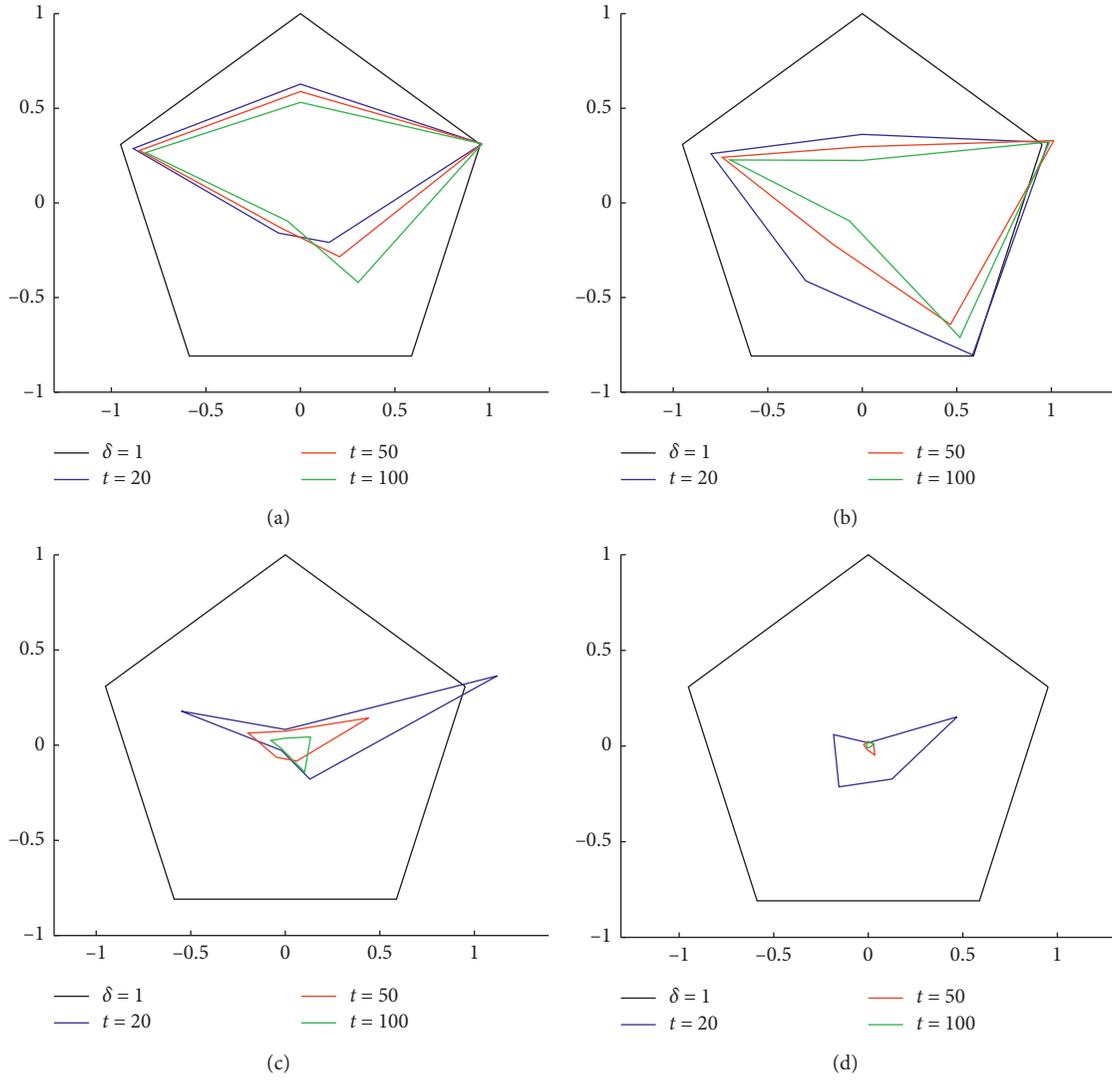


FIGURE 3: The relative errors of each element in the parameter vector. (a) SG. (b) TI-SG. (c) MT-SG. (d) MMG.

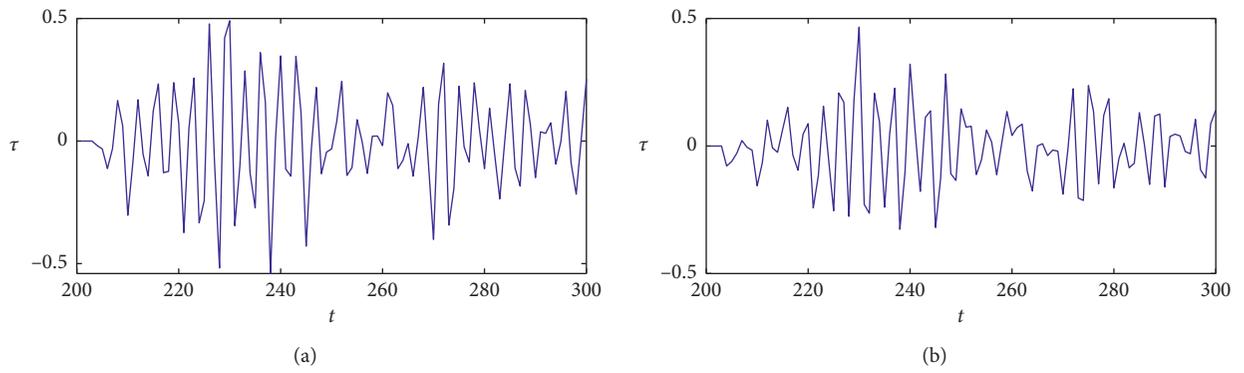


FIGURE 4: Continued.

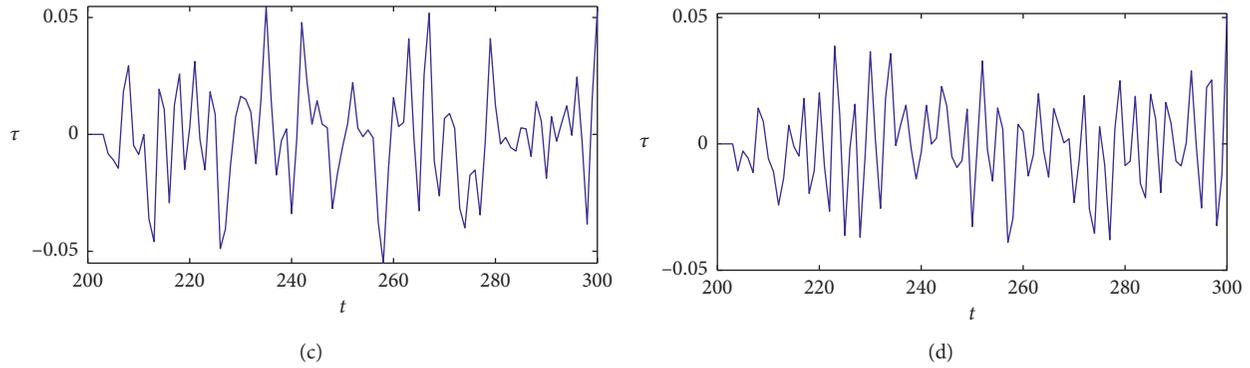


FIGURE 4: Output estimation errors of the four algorithms. (a) SG. (b) TI-SG. (c) MT-SG. (d) MMG.

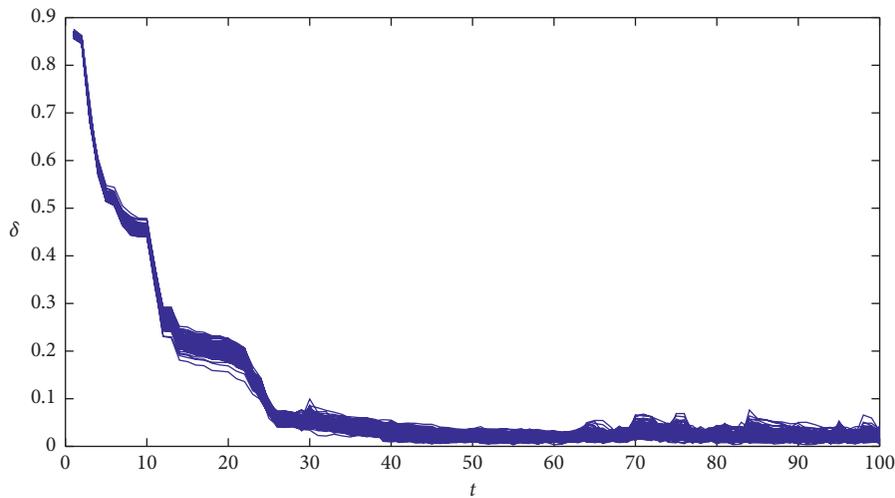


FIGURE 5: The parameter estimation errors by using the Monte Carlo method.

6. Conclusions

This paper proposes an improved gradient descent algorithm for ARX models based on the conjugate gradient descent method. Since two gradient directions and the two corresponding step sizes are involved in each iteration, the proposed algorithm has a quicker convergence rate. The simulation example shows the effectiveness of the proposed algorithm. This algorithm can increase the convergence rate and does not require root calculation. Therefore, it can combine other identification techniques [43–46] to study the parameter estimation issues of linear and nonlinear stochastic systems with colored noises [47–50] and can be extended to other literatures [51–54], such as signal modeling, parameter identification information processing, and engineering application systems [55–57].

Although the MMG algorithm is hoped to be a powerful tool for parameter identification, its convergence property is an open and challenging problem.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61973137), 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).

References

- [1] F. Ding, “Coupled-least-squares identification for multivariable systems,” *IET Control Theory & Applications*, vol. 7, no. 1, pp. 68–79, 2013.
- [2] F. Ding, G. Liu, and X. P. Liu, “Partially coupled stochastic gradient identification methods for non-uniformly sampled systems,” *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1976–1981, 2010.
- [3] F. Ding, Y. Wang, and J. Ding, “Recursive least squares parameter identification algorithms for systems with colored noise using the filtering technique and the auxiliary model,” *Digital Signal Processing*, vol. 37, pp. 100–108, 2015.

- [4] Y. Wang and F. Ding, "Novel data filtering based parameter identification for multiple-input multiple-output systems using the auxiliary model," *Automatica*, vol. 71, pp. 308–313, 2016.
- [5] J. Ding, F. Ding, X. P. Liu, and G. Liu, "Hierarchical least squares identification for linear SISO systems with dual-rate sampled-data," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2677–2683, 2011.
- [6] F. Ding, "Hierarchical multi-innovation stochastic gradient algorithm for Hammerstein nonlinear system modeling," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1694–1704, 2013.
- [7] F. Ding, *System Identification-Iterative Search Principle and Identification Methods*, Science Press, Beijing, China, 2018.
- [8] I. Naveros, C. Ghiaus, D. P. Ruiz, and S. Castaño, "Physical parameters identification of walls using ARX models obtained by deduction," *Energy and Buildings*, vol. 108, no. 12, pp. 317–329, 2015.
- [9] Y. Qin, H. Peng, F. Zhou, X. Zeng, and J. Wu, "Nonlinear modeling and control approach to magnetic levitation ball system using functional weight RBF network-based state-dependent ARX model," *Journal of the Franklin Institute*, vol. 352, no. 10, pp. 4309–4338, 2015.
- [10] R. Haddouche, B. Chetate, and M. S. Boumedine, "Neural network ARX model for gas conditioning tower," *International Journal of Modelling and Simulation*, vol. 39, no. 3, pp. 166–177, 2019.
- [11] J. Na, Z. Yang, S. Kamal, L. Hu, W. Wang, and Y. Zhou, "Bio-inspired learning and adaptation for optimization and control of complex systems," *Complexity*, vol. 2019, Article ID 9325364, 3 pages, 2019.
- [12] J. Na, Y. P. Li, Y. B. Huang, G. Gao, and Q. Chen, "Output feedback control of uncertain hydraulic servo systems," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 490–500, 2019.
- [13] J. Na, B. Jing, Y. Huang, G. Gao, and C. Zhang, "Unknown system dynamics estimator for motion control of nonlinear robotic systems," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 3850–3859, 2020.
- [14] Q. M. Zhu, "A back propagation algorithm to estimate the parameters of non-linear dynamic rational models," *Applied Mathematical Modelling*, vol. 27, no. 3, pp. 169–187, 2003.
- [15] Q. Zhu, D. Yu, and D. Zhao, "An enhanced linear Kalman filter (EnLKF) algorithm for parameter estimation of non-linear rational models," *International Journal of Systems Science*, vol. 48, no. 3, pp. 451–461, 2017.
- [16] D. Wang, L. Mao, and F. Ding, "Recasted models-based hierarchical extended stochastic gradient method for MIMO nonlinear systems," *IET Control Theory & Applications*, vol. 11, no. 4, pp. 476–485, 2017.
- [17] D. Wang, S. Zhang, M. Gan, and J. Qiu, "A novel EM identification method for Hammerstein systems with missing output data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2500–2508, 2020.
- [18] J. Zhang, Q. Zhu, and Y. Li, "Convergence time calculation for supertwisting algorithm and application for nonaffine nonlinear systems," *Complexity*, vol. 2019, Article ID 6235190, 15 pages, 2019.
- [19] J. Zhang, Q. Zhu, Y. Li, and X. L. Wu, "Homeomorphism mapping based neural networks for finite time constraint control of a class of nonaffine pure-feedback nonlinear systems," *Complexity*, vol. 2019, Article ID 9053858, 11 pages, 2019.
- [20] D. Wang, L. Li, Y. Ji, and Y. Yan, "Model recovery for Hammerstein systems using the auxiliary model based orthogonal matching pursuit method," *Applied Mathematical Modelling*, vol. 54, pp. 537–550, 2018.
- [21] F. Ding, F. Wang, L. Xu, T. Hayat, and A. Alsaedi, "Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique," *IET Control Theory & Applications*, vol. 11, no. 3, pp. 390–400, 2017.
- [22] F. Ding, F. Wang, L. Xu, and M. Wu, "Decomposition based least squares iterative identification algorithm for multivariate pseudo-linear ARMA systems using the data filtering," *Journal of the Franklin Institute*, vol. 354, no. 3, pp. 1321–1339, 2017.
- [23] Y. Ji, X. Jiang, and L. Wan, "Hierarchical least squares parameter estimation algorithm for two-input Hammerstein finite impulse response systems," *Journal of the Franklin Institute*, vol. 357, no. 8, pp. 5019–5032, 2020.
- [24] M. H. Li and X. M. Liu, "Maximum likelihood least squares based iterative estimation for a class of bilinear systems using the data filtering technique," *International Journal of Control Automation and Systems*, vol. 18, no. 6, pp. 1581–1592, 2020.
- [25] F. Ding, "Two-stage least squares based iterative estimation algorithm for CARARMA system modeling," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 4798–4808, 2013.
- [26] F. Ding, "Decomposition based fast least squares algorithm for output error systems," *Signal Processing*, vol. 93, no. 5, pp. 1235–1242, 2013.
- [27] A. Jentzen and P. V. Wurstemberger, "Lower error bounds for the stochastic gradient descent optimization algorithm: sharp convergence rates for slowly and fast decaying learning rates," *Journal of Complexity*, vol. 57, Article ID 101438, 2019.
- [28] D. Wang, Y. Yan, Y. Liu, and J. Ding, "Model recovery for Hammerstein systems using the hierarchical orthogonal matching pursuit method," *Journal of Computational and Applied Mathematics*, vol. 345, pp. 135–145, 2019.
- [29] F. Ding and T. Chen, "Performance analysis of multi-innovation gradient type identification methods," *Automatica*, vol. 43, no. 1, pp. 1–14, 2007.
- [30] F. Ding, X. P. Liu, and G. Liu, "Multi-innovation least squares identification for linear and pseudo-linear regression models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 3, pp. 767–778, 2010.
- [31] O. Kyohei, F. Susumu, and S. Shota, "Impact of novel incorporation of CT-based segment mapping into a conjugated gradient algorithm on bone SPECT imaging: fundamental characteristics of a context-specific reconstruction method," *Asia Oceania Journal of Nuclear Medicine & Biology*, vol. 7, no. 1, pp. 49–57, 2019.
- [32] I. S. Armstrong, "Spatial dependence of activity concentration recovery for a conjugate gradient (Siemens xSPECT) algorithm using manufacturer-defined reconstruction presets," *Nuclear Medicine Communications*, vol. 40, no. 3, pp. 287–293, 2019.
- [33] A. R. Heravi and G. A. Hodtani, "A new correntropy-based conjugate gradient backpropagation algorithm for improving training in neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 6252–6263, 2019.
- [34] J. Hu and C. Ma, "Conjugate gradient least squares algorithm for solving the generalized coupled Sylvester-conjugate matrix equations," *Applied Mathematics and Computation*, vol. 334, pp. 174–191, 2018.
- [35] F. Ding, Y. Liu, and B. Bao, "Gradient-based and least-squares-based iterative estimation algorithms for multi-input multi-output systems," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 226, no. 1, pp. 43–55, 2012.

- [36] F. Ding, X. Liu, and J. Chu, "Gradient-based and least-squares-based iterative algorithms for Hammerstein systems using the hierarchical identification principle," *IET Control Theory & Applications*, vol. 7, no. 2, pp. 176–184, 2013.
- [37] L. Xu, L. Chen, and W. Xiong, "Parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration," *Nonlinear Dynamics*, vol. 79, no. 3, pp. 2155–2163, 2015.
- [38] L. Xu, "The damping iterative parameter identification method for dynamical systems based on the sine signal measurement," *Signal Processing*, vol. 120, pp. 660–667, 2016.
- [39] L. Xu, "The parameter estimation algorithms based on the dynamical response measurement data," *Advances in Mechanical Engineering*, vol. 9, no. 11, 2017.
- [40] L. Xu, W. Xiong, A. Alsaedi, and T. Hayat, "Hierarchical parameter estimation for the frequency response based on the dynamical window data," *International Journal of Control, Automation and Systems*, vol. 16, no. 4, pp. 1756–1764, 2018.
- [41] F. Ding, "Combined state and least squares parameter estimation algorithms for dynamic systems," *Applied Mathematical Modelling*, vol. 38, no. 1, pp. 403–412, 2014.
- [42] X. Zhang and F. Ding, "Hierarchical parameter and state estimation for bilinear systems," *International Journal of Systems Science*, vol. 51, no. 2, pp. 275–290, 2020.
- [43] J. Pan, X. Jiang, X. Wan, and W. Ding, "A filtering based multi-innovation extended stochastic gradient algorithm for multivariable control systems," *International Journal of Control, Automation and Systems*, vol. 15, no. 3, pp. 1189–1197, 2017.
- [44] H. Ma, J. Pan, F. Ding, L. Xu, and W. Ding, "Partially-coupled least squares based iterative parameter estimation for multivariable output-error-like autoregressive moving average systems," *IET Control Theory & Applications*, vol. 13, no. 18, pp. 3040–3051, 2019.
- [45] M. Li, X. Liu, and F. Ding, "The filtering-based maximum likelihood iterative estimation algorithms for a special class of nonlinear systems with autoregressive moving average noise using the hierarchical identification principle," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 7, pp. 1189–1211, 2019.
- [46] Y. Zhang, X. Li, G. Zhao, B. Lu, and C. C. Cavalcante, "Signal reconstruction of compressed sensing based on alternating direction method of multipliers," *Circuits, Systems, and Signal Processing*, vol. 39, no. 1, pp. 307–323, 2020.
- [47] J. Pan, W. Li, and H. Zhang, "Control algorithms of magnetic suspension systems based on the improved double exponential reaching law of sliding mode control," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2878–2887, 2018.
- [48] Y. Chang, G. Zhai, B. Fu, and L. Xiong, "Quadratic stabilization of switched uncertain linear systems: a convex combination approach," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 5, pp. 1116–1126, 2019.
- [49] L. Tang, G. J. Liu, M. Yang, F. Li, F. Ye, and C. Li, "Joint design and torque feedback experiment of rehabilitation robot," *Advances in Mechanical Engineering*, vol. 12, no. 5, 2020.
- [50] T. Wu, F. Ye, Y. Su, Y. Wang, and S. Riffat, "Coordinated control strategy of DC microgrid with hybrid energy storage system to smooth power output fluctuation," *International Journal of Low-Carbon Technologies*, vol. 15, no. 1, pp. 46–54, 2020.
- [51] Y. Zhang, M. Huang, T. Wu, and F. Ji, "Reconfigurable equilibrium circuit with additional power supply," *International Journal of Low-Carbon Technologies*, vol. 15, no. 1, pp. 106–111, 2020.
- [52] L. Wang, H. Liu, L. Dai, and Y. Liu, "Novel method for identifying fault location of mixed lines," *Energies*, vol. 11, no. 6, p. 1529, 2018.
- [53] H. Liu, Q. Zou, and Z. Zhang, "Energy disaggregation of appliances consumptions using ham approach," *IEEE Access*, vol. 7, pp. 185977–185990, 2019.
- [54] N. Zhao, Y.-C. Liang, and Y. Pei, "Dynamic contract incentive mechanism for cooperative wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10970–10982, 2018.
- [55] X. Zhao, Z. Lin, B. Fu, L. He, and N. Fang, "Research on automatic generation control with wind power participation based on predictive optimal 2-degree-of-freedom PID strategy for multi-area interconnected power system," *Energies*, vol. 11, no. 12, p. 3325, 2018.
- [56] F. Ding, X. Zhang, and L. Xu, "The innovation algorithms for multivariable state-space models," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 11, pp. 1601–1618, 2019.
- [57] F. Ding, L. Lv, J. Pan, X. Wan, and X.-B. Jin, "Two-stage gradient-based iterative estimation methods for controlled autoregressive systems using the measurement data," *International Journal of Control, Automation and Systems*, vol. 18, no. 4, pp. 886–896, 2020.