Hindawi
Complexity
Volume 2022, Article ID 9244890, 10 pages
https://doi.org/10.1155/2022/9244890

Research Article

Exhaustive Search and Power-Based Gradient Descent Algorithms for Time-Delayed FIR Models

Hua Chen and Yuejiang Ji

Wuxi Vocational College of Science and Technology, Wuxi 214122, China

Correspondence should be addressed to Yuejiang Ji; jyj1981917@126.com

Received 17 April 2022; Revised 16 May 2022; Accepted 27 June 2022; Published 8 September 2022

Academic Editor: Daniele Salvati

In this study, two modified gradient descent (GD) algorithms are proposed for time-delayed models. To estimate the parameters and time-delay simultaneously, a redundant rule method is introduced, which turns the time-delayed model into an augmented model. Then, two GD algorithms can be used to identify the time-delayed model. Compared with the traditional GD algorithms, these two modified GD algorithms have the following advantages: (1) avoid a high-order matrix eigenvalue calculation, thus, are more efficient for large-scale systems; (2) have faster convergence rates, therefore, are more practical in engineering practices. The convergence properties and simulation examples are presented to illustrate the efficiency of the two algorithms.

1. Introduction

System identification plays an important role in control theory and application [1–3]. When the model of a dynamic system is established, one can design robust controllers for such a model to predict its dynamics in the future. There exist many identification algorithms, for example, the least squares (LS) algorithm [4, 5], the gradient descent (GD) algorithm [6, 7], and the particle swarm optimization (PSO) algorithm [8, 9]. When the considered model has a high order, the LS algorithm and the PSO algorithm are inefficient for their heavy computational efforts [10–12]. The GD algorithm has few computational efforts, but with slow convergence rates [13, 14]. To increase the convergence rate of the GD algorithm, two ways are usually performed: (1) design a more suitable direction [15–17]; (2) calculate a better step size [18, 19]. In [20], the best step size of a GD algorithm is given, which involves the eigenvalue calculation. For a high-order matrix, computing its eigenvalues is challenging. To deal with this problem, plenty of suboptimal step size calculating methods are developed, for example, the stochastic GD algorithm [21, 22], the forgetting factor GD algorithm [18, 19], the projection algorithm, and the steepest GD algorithm [23, 24]. Although these algorithms can increase the convergence rates, they are all sensitive to the considered model. That is, one should design different step sizes for different kinds of models.

Time delay is normal in engineering practices. The data of a dynamic system are usually collected by a sensor and then transmitted via a communication channel; they may encounter time-delay due to network congestion [25, 26]. For the time-delayed model identification, Chen proposed a redundant rule-based off-line algorithm that can estimate the parameters and time delay simultaneously [27]. Since the off-line algorithm cannot update the parameters with newly arrived data, Zhang et al. developed a redundant rule-based recursive LS (RLS) algorithm for bilinear time-delayed systems, the RLS algorithm is an online algorithm [28]. This paper focuses on time-delayed model identification and aims to develop some novel identification algorithms which have fast convergence rates and less computational efforts.

Inspired by the PSO algorithm and the power method [29], we propose two modified GD algorithms for time-delayed models: one is an exhaustive search method which chooses the step size based on the PSO algorithm, and the other is the power-based GD algorithm which computes the step size using power method. These two algorithms can get a better step size in each iteration without eigenvalue

calculation. Therefore, the proposed algorithms have faster convergence rates when compared with the traditional GD algorithm.

This paper is organized as follows: Section 2 describes the time-delayed model and the traditional LS and GD algorithms. In Section 3, two modified GD algorithms are proposed. Section 4 proves the convergence properties of the two algorithms. Section 5 gives two simulation examples. Finally, conclusions are presented in Section 6.

2. Problem Statement

First, some notations are denoted as follows: $\mathbf{I}$ denotes an identity matrix of the appropriate sizes; $\|\mathbf{X}\|$ means the norm of the matrix $\mathbf{X}$ and is written as $\|\mathbf{X}\| := \sqrt{\rho(\mathbf{X}^T\mathbf{X})}$; $\rho(\mathbf{X}^T\mathbf{X})$ stands for the spectral radius of the matrix $\mathbf{X}^T\mathbf{X}$; the superscript T is defined as the matrix transpose; $\lambda_{\max}[\mathbf{M}]$ and $\lambda_{\min}[\mathbf{M}]$ denote the maximum and minimum eigenvalues of a matrix $\mathbf{M}$, respectively.

2.1. Time-Delayed Model. Consider the following time-delayed model,

$$y(t) = \sum_{i=1}^{N} g_i u(t-i-\tau) + v(t), \tag{1}$$

where $y(t)$ and $u(t)$ are the output and input, respectively; $v(t)$ is a Gaussian white noise, and satisfies $v(t) \sim N(0, \delta^2)$; $g_i, \quad i = 1, \ldots, N$ are the unknown parameters need to be estimated; τ is an unknown time delay.

Since the time-delay τ is unknown, the corresponding information vector $[u(t-1-\tau), \ldots, u(t-N-\tau)]$ of $y(t)$ is unavailable which leads to the traditional GD algorithm being impossible. To deal with this dilemma, we use the redundant rule method. Assume that the upper bound of the time delay is M , and this assumption is rational and feasible. For example, when using the RIP protocol in the network, the maximum flop is 16.

Rewrite the time-delayed model as follows:

$$\begin{aligned} y(t) =& g_1 u(t-1) + g_2 u(t-2) + \cdots + g_\tau u(t-\tau) \\ &+ g_{(\tau+1)} u(t-\tau-1) + \cdots + g_{(\tau+N)} u(t-\tau-N) \\ &+ g_{(\tau+N+1)} u(t-\tau-N-1) + \cdots \\ &+ g_{(\tau+N+M)} u(t-N-M) + v(t). \end{aligned} \tag{2}$$

Define the parameter vector $\mathbf{G}$ and the information vector ψ as

$$\begin{aligned} \mathbf{G} &= [g_1, \ldots, g_\tau, g_{\tau+1}, \ldots, g_{\tau+N}, g_{\tau+N+1}, \ldots, g_{N+M}]^T \in \mathbb{R}^{M+N}, \\ \boldsymbol{\psi}(t) &= [u(t-1), \ldots, u(t-\tau), u(t-\tau-1), \ldots, u(t-\tau-N), u(t-\tau-N-1), \ldots, u(t-N-M)]^{\mathrm{T}} \in \mathbb{R}^{M+N}. \end{aligned} \tag{3}$$

The augmented parameter vector is decomposed into the following three parts:

$$\begin{aligned} &\text{1. redundant part: } \mathbf{G}_1 = [g_1, \ldots, g_\tau]^{\mathrm{T}}, \\ &\text{2. true part: } \mathbf{G}_2 = [g_{\tau+1}, \ldots, g_{\tau+N}]^{\mathrm{T}}, \\ &\text{3. redundant part: } \mathbf{G}_3 = [g_{\tau+N+1}, \ldots, g_{N+M}]^{\mathrm{T}}. \end{aligned} \tag{4}$$

Remark 1. Since the two corresponding information vectors of the parameter vectors play a less role in the output, the redundant parts $\mathbf{G}_1$ and $\mathbf{G}_2$ are both zero vectors. If the parameter estimates of $\mathbf{G}$ converge to the true values, the two redundant parts equal zero vectors, and then we can obtain the time-delay estimates based on this special structure.

2.2. LS and GD Algorithms. Rewrite the augmented model of the time-delayed model as

$$y(t) = \boldsymbol{\psi}^{\mathrm{T}}(t)\mathbf{G} + v(t). \tag{5}$$

Collect L sets of input and output data and define

$$\begin{aligned} \mathbf{Y}(L) &= [y(L), y(L-1), \ldots, y(1)]^T \in \mathbb{R}^L, \\ \boldsymbol{\Phi}^{\mathrm{T}}(L) &= [\boldsymbol{\psi}(L), \boldsymbol{\psi}(L-1), \ldots, \boldsymbol{\psi}(1)]^T \in \mathbb{R}^{L\times P}, \quad P = (M+N), \\ \mathbf{V}(L) &= [v(L), v(L-1), \ldots, v(1)]^T \in \mathbb{R}^L. \end{aligned} \tag{6}$$

It gives rise to

$$\mathbf{Y}(L) = \boldsymbol{\Phi}^T(L)\mathbf{G} + \mathbf{V}(L). \tag{7}$$

Define the cost function as follows:

$$J(\mathbf{G}) = \frac{1}{2}\left\|\mathbf{Y}(L) - \boldsymbol{\Phi}^T(L)\mathbf{G}\right\|^2. \tag{8}$$

Using the LS algorithm to estimate the parameters, it follows that

$$\hat{G} = \left[\boldsymbol{\Phi}(L)\boldsymbol{\Phi}^T(L)\right]^{-1}\boldsymbol{\Phi}(L)\mathbf{Y}(L). \tag{9}$$

The LS algorithm should perform a matrix inverse calculation which may lead to heavy computational efforts, especially for large-scale systems, e.g., P is large.

To avoid the matrix inverse calculation, the traditional GD (T-GD) algorithm is introduced [20],

$$\hat{\hat{G}}_k = \hat{G}_{k-1} + \gamma_k \mathbf{\Phi}(L)\left[\mathbf{Y}(L) - \mathbf{\Phi}^{\mathrm{T}}(L)\mathbf{G}_{k-1}\right],$$
$$0 < \gamma_k < \frac{2}{\lambda_{\max}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right]}. \tag{10}$$

The T-GD algorithm does not need to compute the inverse of the information matrix $[\Phi(L)\Phi^T(L)]$, but it requires calculating the eigenvalues of the information matrix to choose a suitable step size to keep the T-GD algorithm convergent. When $[\Phi(L)\Phi^T(L)]$ has a high order, computing its eigenvalues is also a challenging problem.

3. Two Modified GD Algorithms

In this section, two modified GD algorithms are developed which aim to avoid eigenvalue calculation and to increase the convergence rate.

3.1. Exhaustive Search-Based GD Algorithm. The PSO algorithm is an intelligent search algorithm, which assigns plenty of particles (initial parameter estimates) first, and then computes the personal best estimates and the global best estimates in each iteration [30, 31]. If the number of the particles is larger, the estimates can easily achieve the true values. Inspired by the PSO algorithm, an exhaustive search-based GD algorithm is developed in this subsection. Its basic idea is to assign several step sizes for a negative direction in each iteration, and the smallest cost function has the best step size.

Assume that the parameter estimate in the iteration $(k-1)$ is $\hat{G}_{k-1}$, the parameter estimate in the iteration k is computed by

$$\hat{G}_k = \hat{G}_{k-1} + \gamma \mathbf{\Phi}(L)\left[\mathbf{Y}(L) - \mathbf{\Phi}^T(L)\mathbf{G}_{k-1}\right]. \tag{11}$$

If we assign a random step size for the above GD algorithm, we can find that (1) a small step size will have a slow convergence rate; (2) a large step size may lead to divergence of the GD algorithm. To choose a suitable step size and to avoid the eigenvalue calculation, we assign S step sizes for the GD algorithm in each iteration.

Define an interval in an iteration k as

$$T_k = [0, S_k]. \tag{12}$$

Choose S uniformly distributed terms between $[0, S_k]$, that is

$$\gamma_k^1 = \frac{S_k}{S}, \cdots, \gamma_k^i = i\frac{S_k}{S}, \cdots, \gamma_k^S = S_k. \tag{13}$$

Based on the S step sizes, we have S parameter estimates, that is

$$\begin{aligned}
\hat{\hat{G}}_k^1 &= \hat{G}_{k-1} + \gamma_k^1 \mathbf{\Phi}(L)\left[\mathbf{Y}(L) - \mathbf{\Phi}^T(L)\mathbf{G}_{k-1}\right] \\
&\vdots \\
\hat{\hat{G}}_k^s &= \hat{G}_{k-1} + \gamma_k^S \mathbf{\Phi}(L)\left[\mathbf{Y}(L) - \mathbf{\Phi}^T(L)\mathbf{G}_{k-1}\right].
\end{aligned} \tag{14}$$

Among the S parameter estimates, we will choose the best one. Once the S parameter estimates in iteration k have been obtained, the S corresponding cost functions are computed by

$$\begin{aligned}
J\left(\hat{\hat{G}}_k^1\right) &= \frac{1}{2}\left\|\mathbf{Y}(L) - \mathbf{\Phi}^T(L)\hat{\hat{G}}_k^1\right\|^2 \\
&\vdots \\
J\left(\hat{\hat{G}}_k^S\right) &= \frac{1}{2}\left\|\mathbf{Y}(L) - \mathbf{\Phi}^T(L)\hat{\hat{\mathbf{G}}}_k^S\right\|^2.
\end{aligned} \tag{15}$$

Let

$$\hat{\hat{G}}_k^{\text{best}} = \arg\min_{\hat{\hat{G}}_k^i}\left[J\left(\hat{\hat{G}}_k^1\right), \ldots, J\left(\hat{\hat{G}}_k^S\right)\right]. \tag{16}$$

That is, the smallest cost function has the best parameter estimate in iteration k.

Then, the steps of the exhaustive search-based GD (ES-GD) algorithm are listed as follows:

Remark 2. The same as the PSO algorithm, a larger S can lead to a more accurate parameter estimate $\hat{G}_k$. However, two problems exist for a poor S_k: (1) if S_k is small, all the step sizes can make $J(\hat{G}_k) < J(\hat{G}_{k-1})$, $i = 1, 2, \ldots, S$, in this case, the step.size is quite small, we then assign $S_k^{\text{new}} = 2S_k^{\text{old}}$; (2) if S_k is too large, all the step sizes lead to $J(\hat{G}_k) > J(\hat{G}_{k-1})$, $i = 1, 2, \ldots, S$, in this case, we should assign $S_k^{\text{new}} = 1/2S_k^{\text{old}}$ to keep the ES-GD algorithm convergent.

Remark 3. The ES-GD algorithm uses the exhaustive search method to choose the step size; the "best" step size in each iteration is better than the step size which is randomly chosen. However, we have no confidence in the "best" step size because it is not the best one. In addition, a larger S can make the "best" step size closer to the true one, but a larger S also leads to heavier computational efforts.

3.2. Power-Based GD Algorithm. In [20], the authors have given the best step size for the cost function

$$J(\mathbf{G}) = \frac{1}{2}\left\|\mathbf{Y}(L) - \mathbf{\Phi}^T(L)\mathbf{G}\right\|^2, \tag{17}$$

is

$$\gamma_{\text{best}} = \frac{2}{\lambda_{\max}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right] + \lambda_{\min}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right]}. \tag{18}$$

Therefore, to get the actual best step size γ_{best}, one should compute both the maximum and minimum eigenvalues of the information matrix $[\Phi(L)\Phi^T(L)]$.

Since the eigenvalues of a high-order matrix are difficult to compute, next, we introduce the power method. The power method can get the maximum eigenvalue of a matrix using an iterative method.

For simplicity, let

Initialise $\hat{G}_0 = \mathbf{1}/p_0$, $p_0 = 10^6$, 1 is a vector whose entries all equal to 1
Collect measurable data $u(1), \ldots, u(L)$ and $y(1), \ldots, y(L)$
Assign the value for S
repeat
  for $k = 1, 2, \cdots$, do
    Assign $[0, S_k]$,
    Choose $\gamma_k^1, \ldots, \gamma_k^S$
    Update $\hat{G}_k^i$, $i = 1, \ldots, S$
    Compute $J(\hat{G}_k^i)$, $i = 1, \ldots, S$
    Compare $J(\hat{G}_k^i)$, $i = 1, \ldots, S$ and choose $\hat{G}_k^{\text{best}}$
    Let $\hat{G}_k = \hat{G}_k^{\text{best}}$
  end
until convergence

Algorithm 1: ES-GD algorithm.

$$\Phi = \left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right]. \tag{19}$$

Assign an initial non-zero vector $\mathbf{x}_0$, and use the following iterative function to get a sequence $\{\mathbf{x}_k\}$,

$$\mathbf{x}_k = \Phi\mathbf{x}_{k-1}. \tag{20}$$

Let

$$\mathbf{x}_k = \left[x_k^1, \ldots, x_k^M\right]^T. \tag{21}$$

The following lemma is obtained.

Lemma 1. *For a symmetric positive definite (SPD) matrix Φ, the sequence $\{\mathbf{x}_k\}$ is computed by (4). Then, the maximum eigenvalue of Φ is computed by*

$$\lambda_{\max}[\Phi] = \frac{x_k^i}{x_{k-1}^i}, \quad i = 1, \text{or}, 2, 3, \ldots, M. \tag{22}$$

Proof. Since Φ is SPD, it has M eigenvalues $\lambda_1, \ldots, \lambda_M$, and their corresponding eigenvectors are $d_1, \ldots, d_2$. Let

$$\|d_i\| = 1, \quad i = 1, 2, \ldots, M, \tag{23}$$

and $d_1, d_2, \ldots, d_M$ are linearly independent. There exist M constants $\alpha_1, \alpha_2, \ldots, \alpha_M$ which are not all equal to zero, and the initial vector $\mathbf{x}_0$ can be written by

$$\mathbf{x}_0 = \sum_{i=1}^{M} \alpha_i d_i. \tag{24}$$

Without loss of generality, assume that the eigenvalues of Φ satisfy

$$\lambda_1 \rangle \lambda_2 \rangle \cdots \rangle \lambda_M \rangle 0. \tag{25}$$

Based on (20), it gives rise to

$$\begin{aligned} \mathbf{x}_k &= \Phi\mathbf{x}_{k-1} \\ &= \Phi^k\mathbf{x}_0 \\ &= \sum_{i=1}^{M} \alpha_i \Phi^k d_i \\ &= \sum_{i=1}^{M} \alpha_i \lambda_i^k d_i. \end{aligned} \tag{26}$$

It follows that

$$\mathbf{x}_k = \lambda_1^k \left[\alpha_1 d_1 + \sum_{i=2}^{M} \left(\frac{\lambda_i}{\lambda_1}\right)^k \alpha_i d_i\right]. \tag{27}$$

Since $\lambda_1 \rangle \lambda_i$, $i = 2, 3, \ldots, M$, when $k \longrightarrow \infty$, we have

$$\lim_{k \longrightarrow \infty} \left(\frac{\lambda_i}{\lambda_1}\right)^k = 0. \tag{28}$$

Then, (27) can be rewritten by

$$\mathbf{x}_k = \lambda_1^k \alpha_1 d_1. \tag{29}$$

Therefore, we can get that

$$\lambda_1 = \frac{x_k^i}{\left(x_k^i - 1\right)}, \quad k \longrightarrow \infty. \tag{30}$$

The proof is completed.

The power method can only get the maximum eigenvalue of Φ. However, to get the best step size, one also should compute the minimum eigenvalue of Φ. Next, we introduce an effective method to compute the minimum eigenvalue of Φ.

Once the maximum eigenvalue of Φ is obtained, we then assign a new term $\overline{\lambda}$ as follows:

$$\overline{\lambda} = (\lambda_1 + \delta), \delta \rangle 0, \tag{31}$$

where δ is a positive constant which is chosen on a case by case basis.

Define a new matrix Ψ

$$\Psi = \overline{\lambda}\mathbf{I} - \Phi. \tag{32}$$

Then, the following lemma can be obtained. □

Lemma 2. *For a symmetric positive definite matrix Φ , its eigenvalues are $\lambda_1 \rangle \lambda_2 \rangle \cdots \rangle \lambda_M \rangle 0$. A matrix Ψ is defined by (6). Then, the eigenvalues of the matrix Ψ are*

$$\begin{aligned} \lambda_i^{\Psi} &= \overline{\lambda} - \lambda_i,\ i = 1, 2, \ldots, M, \\ 0 &\langle \lambda_1^{\Psi} \langle \lambda_2^{\Psi} \langle \cdots \langle \lambda_M^{\Psi}. \end{aligned} \tag{33}$$

Proof. For an SPD matrix Φ, there exists a nonsingular matrix $\mathbf{Q}$ which can guarantee

$$\Phi = \mathbf{Q}^{-1}\text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_M\}\mathbf{Q}. \tag{34}$$

Then, the matrix Ψ is written by

$$\begin{aligned} \Psi &= \mathbf{Q}^{-1}\text{diag}\left\{\overline{\lambda}, \overline{\lambda}, \ldots, \overline{\lambda}\right\}\mathbf{Q} - \mathbf{Q}^{-1}\text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_M\}\mathbf{Q} \\ &= \mathbf{Q}^{-1}\text{diag}\left\{\overline{\lambda} - \lambda_1, \overline{\lambda} - \lambda_2, \ldots, \overline{\lambda} - \lambda_M\right\}\mathbf{Q} \\ &= \mathbf{Q}^{-1}\text{diag}\left\{\lambda_1^{\Psi}, \lambda_2^{\Psi}, \ldots, \lambda_M^{\Psi}\right\}\mathbf{Q}. \end{aligned} \tag{35}$$

Since $\overline{\lambda} = (\lambda_1 + \delta)$ and $\delta \rangle 0$, we have

$$0 \langle \lambda_1^{\Psi} \langle \lambda_2^{\Psi} \langle \cdots \langle \lambda_M^{\Psi}. \tag{36}$$

For the SPD matrix Ψ, using the power method can obtain its maximum eigenvalue λ_M^{Ψ}, and then the minimum eigenvalue λ_M of the matrix Φ can be computed by

$$\lambda_M = \overline{\lambda} - \lambda_M^{\Psi}. \tag{37}$$

When the maximum and minimum eigenvalues of the matrix Φ are obtained, we can get the best step size.

The steps of the power-based GD (P-GD) algorithm are listed as follows: □

Remark 4. If the maximum eigenvalue λ_1 is not much bigger than λ_2, to compute the maximum eigenvalue λ_1 is time-consuming. Because the value of $[\lambda_2/\lambda_1]^k$ will take more iterations to converge to zero.

Remark 5. The choice of the positive constant δ is very important; to get the maximum λ_M^{Ψ} quickly, we would do better to choose a small δ. On the other hand, due to the estimation error, a small δ may lead Ψ not be an SPD matrix.

Remark 6. Recently, a novel GD algorithm, termed as fractional stochastic GD algorithm, has been proposed for parameter estimation. This algorithm is a well complement to the traditional GD algorithm, which can be widely used for different kinds of models [32–34].

4. Convergence Properties of the Two Modified GD Algorithms

The convergence properties of the two modified GD algorithms are given in the following which offer theory guidance for researchers.

4.1. Convergence Analysis of the ES-GD Algorithm. Rewrite the ES-GD algorithm as follows:

$$\hat{G}_k = \hat{G}_{k-1} + \gamma\mathbf{\Phi}(L)\left[\mathbf{Y}(L) - \mathbf{\Phi}^T(L)\mathbf{G}_{k-1}\right]. \tag{38}$$

Subtracting $\mathbf{G}$ on both sides of the above equation yields

$$\begin{aligned} \hat{E}_k &= \hat{E}_{k-1} - \gamma\mathbf{\Phi}(L)\left[\mathbf{\Phi}^T(L)\mathbf{E}_{k-1}\right] + \gamma\mathbf{\Phi}(L)\mathbf{V}(L) \\ &= \left[\mathbf{I} - \gamma\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right]\mathbf{E}_{k-1} + \gamma\mathbf{\Phi}(L)\mathbf{V}(L), \end{aligned} \tag{39}$$

where $\hat{E}_k = (\hat{G}_k - \mathbf{G})$. Since $\mathbf{V}(L)$ is a Gaussian white noise and is independent on $\Phi(L)$, the above equation is simplified as

$$\hat{E}_k = \left[\mathbf{I} - \gamma\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right]\mathbf{E}_{k-1}. \tag{40}$$

Based on the exhaustive search-based, in each iteration, we will find an optimal γ which guarantees

$$\left\|\mathbf{I} - \gamma\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right\| \le 1. \tag{41}$$

It gives rise to

$$\left\|\hat{E}_k\right\| \le \left\|\mathbf{E}_{k-1}\right\|. \tag{42}$$

Therefore, the ES-GD algorithm is convergent.

4.2. Convergence Analysis of the P-GD Algorithm. The P-GD algorithm is written by

$$\begin{aligned} \hat{G}_k = \hat{G}_{k-1} &+ \frac{2}{\lambda_{\max}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right] + \lambda_{\min}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right]} \\ &\cdot \mathbf{\Phi}(L)\left[\mathbf{Y}(L) - \mathbf{\Phi}^T(L)\mathbf{G}_{k-1}\right]. \end{aligned} \tag{43}$$

Subtracting the true value $\mathbf{G}$ on both sides of the above equation yields

$$\begin{aligned} \hat{E}_k &= \hat{E}_{k-1} - \frac{2}{\lambda_{\max}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right] + \lambda_{\min}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right]} \\ &\quad \cdot \mathbf{\Phi}(L)\left[\mathbf{\Phi}^T(L)\mathbf{E}_{k-1}\right] \\ &= \left[\mathbf{I} - \frac{2\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)}{\lambda_{\max}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right] + \lambda_{\min}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right]}\right]\hat{E}_{k-1}. \end{aligned} \tag{44}$$

Initialise $\hat{G}_0 = 1/p_0$, $p_0 = 10^6$, $\mathbf{1}$ is a vector whose entries all equal to 1
Collect measurable data $u(1), \ldots, u(L)$ and $y(1), \ldots, y(L)$
Use the power method to compute λ_1
Assign a positive constant δ based on λ_1
Construct an SPD matrix Ψ
Use the power method to compute the maximum eigenvalue λ_M^{Ψ} of Ψ
Calculate λ_M based on λ_M^{Ψ}
Compute the best step size γ_{best}
repeat
  for $k = 1, 2, \cdots,$ do
    Update $\hat{G}_k$
  end
until convergence

Algorithm 2: P-GD algorithm.

For simplicity, let

$$\begin{aligned}\lambda_1 &= \lambda_{\max}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right],\\ \lambda_M &= \lambda_{\min}\left[\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)\right].\end{aligned} \tag{45}$$

Equation (44) is simplified as

$$\hat{E}_k = \left[\mathbf{I} - \frac{2\mathbf{\Phi}(L)\mathbf{\Phi}^T(L)}{\lambda_1 + \lambda_M}\right]\hat{E}_{k-1}. \tag{46}$$

For an SPD matrix $\Phi(L)\Phi^T(L)$, there exists a matrix $\mathbf{Q}$ which can ensure

$$\mathbf{\Phi}(L)\mathbf{\Phi}^T(L) = \mathbf{Q}^{-1}\text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_M]\mathbf{Q}. \tag{47}$$

It follows that equation (46) can be transformed into

$$\begin{aligned}\hat{E}_k &= \left[\mathbf{Q}^{-1}\mathbf{Q} - \frac{2}{(\lambda_1 + \lambda_M)}\mathbf{Q}^{-1}\text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_M]\mathbf{Q}\right]\hat{E}_{k-1}\\ &= \left[\mathbf{Q}^{-1}\text{diag}\left[1 - \frac{\lambda_1}{(\lambda_1 + \lambda_M)}, 1 - \frac{\lambda_2}{(\lambda_1 + \lambda_M)}, \ldots, 1 - \frac{\lambda_M}{(\lambda_1 + \lambda_M)}\right]\mathbf{Q}\right]\hat{E}_{k-1}\\ &= \left[\mathbf{Q}^{-1}\text{diag}\left[\frac{(\lambda_M - \lambda_1)}{(\lambda_1 + \lambda_M)}, \ldots, \frac{(\lambda_1 - \lambda_M)}{(\lambda_1 + \lambda_M)}\right]\mathbf{Q}\right]\hat{E}_{k-1}.\end{aligned} \tag{48}$$

Clearly, all the absolute values in the diagonal matrix are smaller than 1, then we have

$$\left\|\hat{E}_k\right\| \leq \left\|\hat{E}_{k-1}\right\|. \tag{49}$$

Therefore, the P-GD algorithm is convergent.

Remark 7. In the P-GD algorithm, the maximum absolute value in the diagonal matrix is $(\lambda_1 - \lambda_M)/(\lambda_1 + \lambda_M)$, that is,

$$\left\|\hat{E}_k\right\| = \frac{(\lambda_1 - \lambda_M)}{(\lambda_1 + \lambda_M)}\left\|\hat{E}_{k-1}\right\| = \frac{(\tau - 1)}{(\tau + 1)}\left\|\hat{E}_{k-1}\right\|, \tag{50}$$

where $\tau = \lambda_1/\lambda_M$ is the conditioned number of the matrix $[\Phi(L)\Phi^T(L)]$. If the matrix $[\Phi(L)\Phi^T(L)]$ is ill-conditioned, no matter what the step size is, the convergence rates are always very slow. In this case, we can try to reconstruct a new information matrix $[\Phi(L)\Phi^T(L)]$ or use the fractional stochastic GD algorithm proposed in [32–34] to increase the convergence rates.

5. Examples

Example 1. Consider the following time-delayed model,

$$\begin{aligned}y(t) &= g_1u(t-\tau-1) + g_2u(t-\tau-2) + g_3u(t-\tau-3)\\ &\quad + g_4u(t-\tau-4) + g_5u(t-\tau-5) + v(t)\\ &= 0.5u(t-\tau-1) + 0.67u(t-\tau-2)\\ &\quad - 0.34u(t-\tau-3) + 0.23u(t-\tau-4)\\ &\quad + 0.76u(t-\tau-5) + v(t).\end{aligned} \tag{51}$$

Assume that the time delay is $\tau = 2$ and assign $M = 5$, we have

$$\begin{aligned}y(t) &= g_{0,1}u(t-1) + g_{0,2}u(t-2) + g_1u(t-3)\\ &\quad + g_2u(t-4) + g_3u(t-5) + g_4u(t-6)\\ &\quad + g_5u(t-7) + g_{1,1}u(t-8) + g_{1,2}u(t-9)\\ &\quad + g_{1,3}u(t-10) + v(t),\\ \mathbf{G} &= [0, 0, 0.5, 0.67, -0.34, 0.23, 0.76, 0, 0, 0]^T.\end{aligned} \tag{52}$$

In simulation, we collect 500 sets of input and output data, where $u(t) \sim N(0,1)$ and $v(t) \sim N(0, 0.1)^2$. Use the T-GD, ES-GD, and P-GD algorithms for the time-delayed model. The parameter estimates and their estimation errors $\varsigma = \|\mathbf{G} - \mathbf{G}^k\|/\mathbf{G}$ are shown in Figure 1 and Table 1. The elapsed times of these three algorithms are illustrated in Table 2: the second row means that all the three algorithms run the same iteration, and the third row shows that the three algorithms have almost the same estimation error.

Assign a threshold $\varrho = 0.005$. Compare the estimates (20-th iteration) with the threshold, if the absolute value of the estimate is smaller than the threshold, then it will be assigned as zero. We can get that the time-delay is 2.

In addition, we use the power method to compute the maximum and minimum eigenvalues of the information

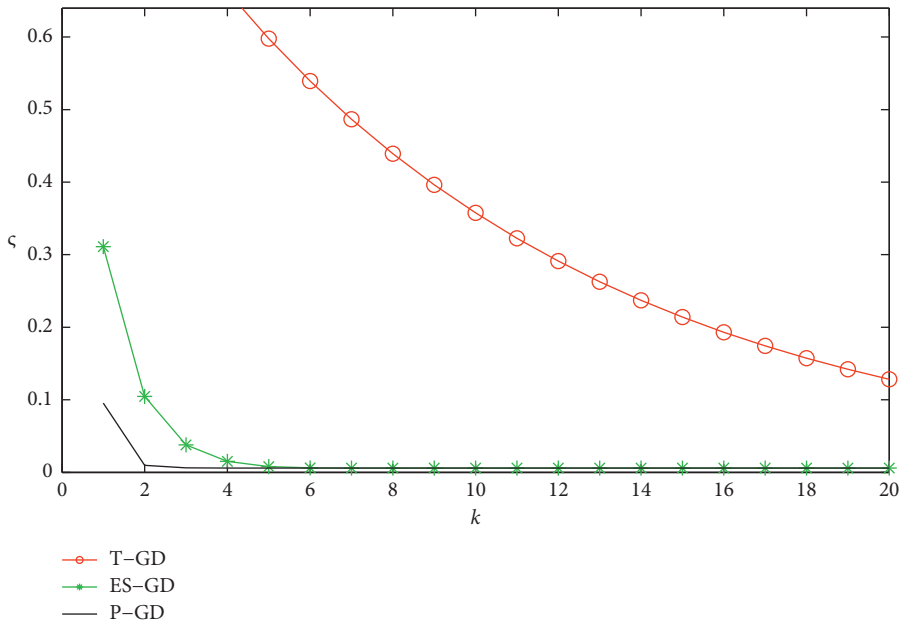

Figure 1: The parameter estimation errors ς versus k.

Table 1: The parameter estimates and their estimation errors.

Algorithms	k	$g_{0,1}$	$g_{0,2}$	g_1	g_2	g_3	g_4	g_5	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$\varsigma(\%)$
T-GD	1	−0.00076	0.00014	0.04923	0.06527	−0.03305	0.02249	0.07435	0.00028	−0.00061	0.00039	90.22459
	2	−0.00139	0.00027	0.09359	0.12417	−0.06288	0.04280	0.14142	0.00050	−0.00111	0.00070	81.40546
	5	−0.00268	0.00062	0.20209	0.26859	−0.13612	0.09259	0.30580	0.00092	−0.00205	0.00134	59.79451
	10	−0.00351	0.00104	0.32227	0.42938	−0.21783	0.14806	0.48858	0.00109	−0.00248	0.00172	35.76318
	15	−0.00354	0.00132	0.39373	0.52564	−0.26688	0.18129	0.59784	0.00096	−0.00227	0.00171	21.39891
	20	−0.00328	0.00150	0.43622	0.58328	−0.29632	0.20121	0.66316	0.00073	−0.00186	0.00157	12.81303
ES-GD	1	0.01755	−0.02912	0.31694	0.47762	−0.26669	0.15867	0.52597	−0.04381	0.01115	0.02908	31.10412
	2	0.01217	−0.01885	0.43294	0.61302	−0.32675	0.20761	0.68377	−0.02656	0.00853	0.01859	10.47646
	5	0.00153	−0.00351	0.49623	0.66886	−0.34469	0.22932	0.75506	−0.00126	0.00332	−0.00035	0.78800
	10	0.00023	−0.00230	0.49948	0.67068	−0.34518	0.23005	0.75812	0.00101	0.00266	−0.00251	0.59350
	15	0.00021	−0.00229	0.49951	0.67069	−0.34519	0.23005	0.75815	0.00103	0.00265	−0.00254	0.59388
	20	0.00021	−0.00229	0.49951	0.67069	−0.34519	0.23005	0.75815	0.00103	0.00265	−0.00254	0.59389
P-GD	1	0.02557	−0.04243	0.46184	0.69598	−0.38862	0.23121	0.76643	−0.06384	0.01625	0.04237	9.53122
	2	0.00246	−0.00123	0.49699	0.66530	−0.33847	0.22943	0.75110	0.00198	0.00327	0.00074	0.96875
	5	0.00022	−0.00229	0.49950	0.67070	−0.34520	0.23005	0.75814	0.00102	0.00266	−0.00253	0.59415
	10	0.00021	−0.00229	0.49951	0.67069	−0.34519	0.23005	0.75815	0.00103	0.00265	−0.00254	0.59389
	15	0.00021	−0.00229	0.49951	0.67069	−0.34519	0.23005	0.75815	0.00103	0.00265	−0.00254	0.59389
	20	0.00021	−0.00229	0.49951	0.67069	−0.34519	0.23005	0.75815	0.00103	0.00265	−0.00254	0.59389
True values		0.00000	0.00000	0.50000	0.67000	−0.34000	0.23000	0.76000	0.00000	0.00000	0.00000	0.00000

Table 2: The elapsed times of the three algorithms.

Algorithm	T-GD	ES-GD	P-GD
Elapsed time (second)	0.2482 ($k=20$)	0.3125 ($k=20$)	0.2876 ($k=20$)
Elapsed time (second)	0.3516 ($k=40$)	0.1325 ($k=6$)	0.1008 ($k=4$)

matrix $[\Phi(L)\Phi^T(L)]$, and the estimates are shown in Figure 2.

From this simulation, we can obtain the following conclusions:

(1) The P-GD algorithm has the fastest convergence rates, then is the ES-GD algorithm, and the T-GD algorithm has the slowest convergence rates; this can be shown in Figure 1;

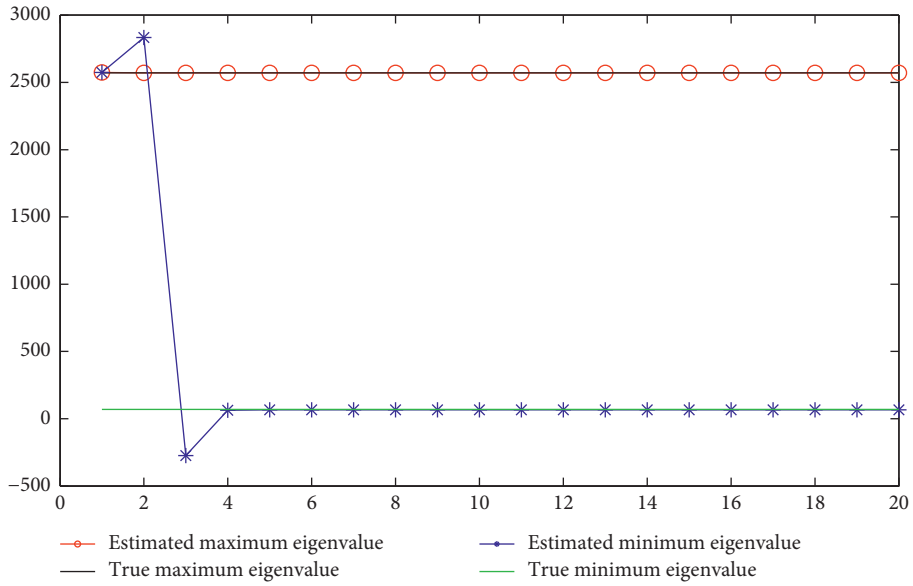

Figure 2: The maximum and minimum eigenvalue estimates versus k.

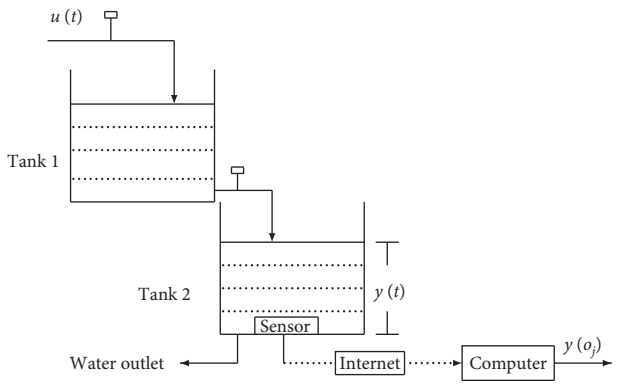

Figure 3: A water tank system.

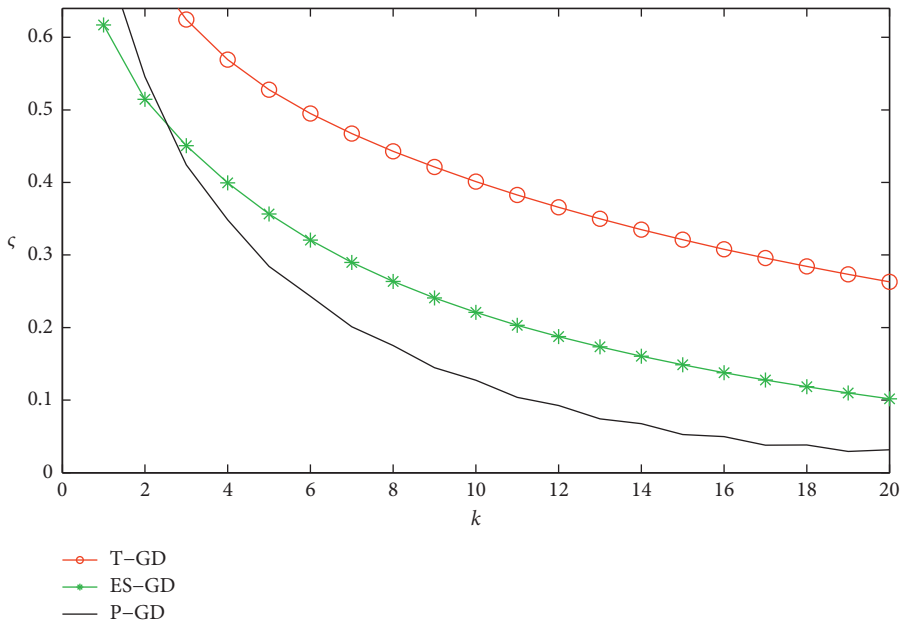

Figure 4: The parameter estimation errors ς versus k.

(2) All the three algorithms can obtain the parameter estimates and the time-delay estimates simultaneously, as shown in Table 1;

(3) The P-GD algorithm is the most effective algorithm, follows the ES-GD algorithm, and the last one is the T-GD algorithm; this can be shown in Table 2;

(4) The power method can obtain the maximum eigenvalue and the minimum eigenvalue of the information matrix, as shown in Figure 2.

Example 2. A water tank system with a communication channel is proposed for simulation, see Figure 3, where $u(t)$ is the position of the inlet water valve, and $y(t)$ is the water level of Tank 2 and sampled by a pressure sensor. There exists a time-delay $\tau = 3$. The water tank system is modeled by the following model [35]:

$$\begin{aligned} y(t) = {} & 0.3y(t-1-3) - 0.1y(t-2-3) \\ & + 0.15y(t-3-3) - 0.3u(t-1) \\ & + 0.2u(t-2) + 0.13u(t-3) + v(t). \end{aligned} \tag{53}$$

Using the T-GD, ES-GD, and P-GD algorithms for this model, the parameter estimates and their estimation errors are shown in Figure 4.

This example also shows that the two modified GD algorithms have faster convergence rates than those of the T-GD algorithm.

6. Conclusions

Two modified GD algorithms are proposed for systems with time-delay in this paper. The first is the ES-GD algorithm that does not require the eigenvalue calculation. The second is the P-GD algorithm, that can get the best step size by using the power method. These two modified GD algorithms have faster convergence rates than those of the T-GD algorithm, and can obtain the parameter estimates and time-delay estimates simultaneously. Thus, they can be widely used in engineering practices.

In this paper, we only use the modified GD algorithms for the time-delayed systems. If the systems have other kinds of hidden variables, e.g., missing outputs and model identities, can these two algorithms be also effective? this topic will remain as an open issue in future.

Data Availability

All data generated or analyzed during this study are included in this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangsu Province (No. BK20131109).

References

[1] T. So derstrom and P. Stoica, *Systen Identification*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

[2] J. Chen, J. Ma, M. Gan, and Q. Zhu, "Multi-direction gradient iterative algorithm: a unified framework for gradient iterative and least squares algorithms," *IEEE Transactions on Automatic Control*, 2021.

[3] J. Chen, B. Huang, M. Gan, and C. P. Chen, "A novel reduced-order algorithm for rational models based on Arnoldi process and Krylov subspace," *Automatica*, vol. 129, Article ID 109663, 2021.

[4] G. Y. Chen, M. Gan, C. L. P. Chen, and H. X. Li, "A regularized variable projection algorithm for separable nonlinear least-squares problems," *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp. 526–537, 2019.

[5] G. Birpoutsoukis, A. Marconato, J. Lataire, and J. Schoukens, "Regularized nonparametric Volterra kernel estimation," *Automatica*, vol. 82, pp. 324–327, 2017.

[6] F. Ding, H. Ma, J. Pan, and E. F. Yang, "Hierarchical gradient- and least squares-based iterative algorithms for input nonlinear output-error systems using the key term separation," *Journal of the Franklin Institute*, vol. 358, no. 9, pp. 5113–5135, 2021.

[7] Y. J. Ji and L. X. Lv, "Two identification methods for a nonlinear membership function," *Complexity*, vol. 2021, Article ID 5515888, 7 pages, 2021.

[8] J. H. Li and X. Li, "Particle swarm optimization iterative identification algorithm and gradient iterative identification algorithm for Wiener systems with colored noise," *Complexity*, vol. 2018, Article ID 7353171, 2018.

[9] J. H. Li, T. C. Zong, J. P. Gu, and L. Hua, "Parameter estimation of Wiener systems based on the particle swarm iteration and gradient search principle," *Circuits, Systems, and Signal Processing*, vol. 39, no. 7, pp. 3470–3495, 2020.

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

[11] Y. Chen, Y. J. Liu, J. Chen, and J. X. Ma, "A novel identification method for a class of closed-loop systems based on basis pursuit de-noising," *IEEE Access*, vol. 8, Article ID 99648, 2020.

[12] 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 Dynamics*, vol. 101, no. 1, pp. 361–373, 2020.

[13] S. J. Fan, L. Xu, F. Ding, A. Alsaedi, and T. Hayat, "Correlation analysis-based stochastic gradient and least squares identification methods for errors-in-variables systems using the multi-innovation," *International Journal of Control, Automation and Systems*, vol. 19, no. 1, pp. 289–300, 2021.

[14] J. Chen, F. Ding, Q. Zhu, and Y. J. Liu, "Interval error correction auxiliary model based gradient iterative algorithms for multirate ARX models," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4385–4392, 2020.

[15] D. Q. Wang, Y. R. Yan, Y. J. Liu, and J. H. 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.

[16] D. Q. Wang, Q. H. Fan, and Y. Ma, "An interactive maximum likelihood estimation method for multivariable Hammerstein systems," *Journal of the Franklin Institute*, vol. 357, no. 17, Article ID 12986, 2020.

[17] M. James, "New insights and perspectives on the natural gradient method," 2014, https://arxiv.org/pdf/1412.1193.

[18] Q. L. Liu, Y. S. Xiao, F. Ding, and T. Hayat, "Decomposition-based over-parameterization forgetting factor stochastic gradient algorithm for Hammerstein-Wiener nonlinear systems with non-uniform sampling," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 12, pp. 6007–6024, 2021.

[19] 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," *Journal of the Franklin Institute*, vol. 353, no. 16, pp. 4280–4299, 2016.

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

[21] C. P. Yu, J. Chen, S. K. Li, and M. Verhaegen, "Identification of affinely parameterized state-space models with unknown inputs," *Automatica*, vol. 122, Article ID 109271, 2020.

[22] 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 Transactions on Cybernetics*, vol. 51, no. 2, pp. 614–623, 2021.

[23] S. Magnusson, C. Enyioha, N. Li, C. Fischione, and V. Tarokh, "Convergence of limited communication gradient methods," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1356–1371, 2018.

[24] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[25] Y. Gu, Q. M. Zhu, C. J. Li, P. Y. Zhu, and H. Nouri, "State filtering and parameter estimation for two-input two-output systems with time delay," *IET Control Theory & Applications*, vol. 15, no. 16, pp. 2053–2066, 2021.

[26] D. Q. Wang, Z. Zhang, and B. Xue, "Decoupled parameter estimation methods for Hammerstein systems by using filtering technique," *IEEE Access*, vol. 6, Article ID 66612, 2018.

[27] J. Chen, J. X. Ma, Y. J. Liu, and F. Ding, "Identification methods for time-delay systems based on the redundant rules," *Signal Processing*, vol. 137, pp. 192–198, 2017.

[28] X. Zhang, Q. Y. Liu, F. Ding, A. Alsaedi, and T. Hayat, "Recursive identification of bilinear time-delay systems through the redundant rule," *Journal of the Franklin Institute*, vol. 357, no. 1, pp. 726–747, 2020.

[29] J. Chen, D. Q. Wang, Y. J. Liu, and Q. M. Zhu, "Varying infimum gradient descent algorithm for agent-sever systems using different order iterative preconditioning methods," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4436–4446, 2022.

[30] Y. Y. Cui, X. Meng, and J. F. Qiao, "A multi-objective particle swarm optimization algorithm based on two-archive mechanism," *Applied Soft Computing*, vol. 119, Article ID 108532, 2022.

[31] P. B. Fernandes, R. C. L. Oliveira, and J. Fonseca Neto, "Trajectory planning of autonomous mobile robots applying a particle swarm optimization algorithm with peaks of diversity," *Applied Soft Computing*, vol. 116, Article ID 108108, 2022.

[32] Z. A. Khan, N. I. Chaudhary, and S. Zubair, "Fractional stochastic gradient descent for recommender systems," *Electronic Markets*, vol. 29, no. 2, pp. 275–285, 2019.

[33] Z. A. Khan, S. Zubair, H. Alquhayz, M. Azeem, and A. Ditta, "Design of momentum fractional stochastic gradient descent for recommender systems," *IEEE Access*, vol. 7, Article ID 179575, 2019.

[34] Z. A. Khan, S. Zubair, N. I. Chaudhary, M. A. Z. Raja, F. A. Khan, and N. Dedovic, "Design of normalized fractional SGD computing paradigm for recommender systems," *Neural Computing & Applications*, vol. 32, no. 14, Article ID 10245, 2020.

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