Hindawi
Complexity
Volume 2023, Article ID 7527478, 12 pages
https://doi.org/10.1155/2023/7527478

*Research Article*

# Fractional Gradient Descent-Based Auxiliary Model Algorithm for FIR Models with Missing Data

**Jia Tang**

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

Correspondence should be addressed to Jia Tang; 1201052@wxsc.edu.cn

Received 18 September 2022; Revised 31 December 2022; Accepted 19 January 2023; Published 9 February 2023

Academic Editor: Abdellatif Ben Makhlouf

This study proposes a fractional gradient descent (FGD) algorithm for FIR models with missing data. By using the auxiliary model method, the missing data can be obtained. Then, the FGD algorithm is applied to update the parameters of the FIR models. Because of the fractional term in the conventional GD algorithm, the convergence rates of the GD algorithm can be increased. In addition, to avoid the step-size calculation, an Aitken FGD-based auxiliary model algorithm is also introduced. The convergence analysis and simulation examples are provided to show the effectiveness of the proposed algorithms.

## 1. Introduction

The gradient descent (GD) algorithm is a classical method which is widely used in system identification and parameter estimation [1–3]. Its basic idea is to assign a random initial parameter estimate and then force the estimate to move to another better estimate [4–6]. Clearly, the direction and step-size are the two factors in the GD algorithm designing. Owing to the zigzagging nature, the GD algorithm has a quite slow convergence rate [7–9]. To increase the convergence rates, one can modify the GD algorithm through two ways: (1) choose a better direction instead of using the negative direction; (2) calculate a suitable step-size.

In the last few decades, there have been lots of modified GD algorithms, e.g., the forgetting factor GD algorithm [10], the multiinnovation GD algorithm [11, 12], the conjugate GD algorithm [13], and the momentum GD algorithm [14]. For example, Fan and Liu proposed a forgetting factor GD algorithm for input nonlinear FIR-MA systems, where the colored noise is transformed into a white noise by using the data filtering method [15]. Chen et al. developed a multidirection GD algorithm for ARX models, such an algorithm establishes a link between the GD and LS algorithms [16]. Xu et al. used a three-stage GD algorithm for exponential autoregressive time-series models [17]. Although these algorithms can increase the convergence rates, they bring other challenging issues: (1) heavy computational efforts, e.g., the multidirection GD algorithm; (2) less confidence of the convergence properties, e.g., the three-stage GD algorithm; and (3) large variance of the parameter estimates, e.g., the forgetting factor GD algorithm.

The FGD algorithm proposed in [18, 19] is an outstanding alternative of the GD algorithm. Its basic idea is to add a fractional gradient direction to the GD algorithm [20–22]. With the help of the additional direction, the convergence rates of the GD algorithm are increased. Recently, a two-stage fractional least mean square identification algorithm is proposed for a large-scale linear system; this large-scale system is decomposed into two parts, and these two parts are iteratively identified [23]. Chaudhary et al. developed two FGD algorithms for signal modeling, where these two algorithms require less computational efforts [24]. Xu et al. provided a momentum-based FGD and an adaptive-based FGD algorithm for time-delayed ARX models; these two algorithms have fast convergence rates and less computational efforts [25]. Khan et al. proposed a fractional SGD algorithm for recommender systems, and this algorithm provides fast and accurate recommendations by using a time-varying learning rate [26]. However, the above work assumed that all the data from the models were available.

Owing to the network congestion, some data of the dynamic systems are missing [27–29]. Usually, two standard tools are applied to systems with missing data: (1) the lifting technique, e.g., the polynomial transformation algorithm [30], whose key is to transform the system into a lifted system with high order and then estimate the parameters based on all the measurable data; (2) the imputation technique, e.g., the auxiliary model method [31], first estimates the missing data and then updates the parameters based on the measurable data and estimated data. Since the lifting method can enlarge the number of unknown parameters, this study focuses on the imputation method.

This study proposes an FGD-based auxiliary model (FGD-AM) algorithm for FIR models with missing data. Based on the auxiliary model method and the FGD algorithm, the missing data and parameters can be iteratively estimated. In addition, to avoid calculating the step-size, an Aitken FGD-AM (A-FGD-AM) algorithm is also proposed. Compared with the traditional algorithms, the method in this study has the following contributions: (1) the method has faster convergence rates; (2) the method does not require calculating the step-size; and (3) the method avoids the matrix inverse calculation.

The study is organized as follows. Section 2 describes the FIR models with missing data. Section 3 proposes the FGD-AM algorithm. Section 4 gives the A-FGD-AM algorithm. Section 5 provides two simulation examples. Finally, conclusions and future directions are presented in Section 6.

## 2. Problem Statement

First, setup some notations. The norm of a matrix $\mathbf{X}$ is defined as $\|\mathbf{X}\| = \sqrt{\lambda_{\max}[\mathbf{X}\mathbf{X}^{\mathrm{T}}]}$; $\lambda_{\max}[\mathbf{X}\mathbf{X}^{\mathrm{T}}]$ and $\lambda_{\min}[\mathbf{X}\mathbf{X}^{\mathrm{T}}]$ mean the maximum and minimum eigenvalues of matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$, respectively; the norm of a vector $\mathbf{z} = [z_1, z_2, \ldots, z_n]^{\mathrm{T}} \in \mathbb{R}^n$ is defined as $\|\mathbf{z}\| = (\sum_{i=1}^{n} z_i^2)^{1/2}$; the superscript $T$ denotes the matrix transpose.

*2.1. FIR Model with Missing Data.* Consider the following FIR model with missing data:

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

where $y(t)$ and $u(t)$ are the system output and input, respectively, $v(t)$ is the noise, and $F(z)$ is a polynomial which is written by

$$F(z) = f_1 z^{-1} + f_2 z^{-2} + \cdots + f_n z^{-n}. \tag{2}$$

With the development of the communication network and sensor technologies, researchers usually use sensors to collect the data and then transmit them over a communication network. Some data are often missed due to the network congestion. In this study, we assume that some output data are missing, the data at the sampling instants $m_1, m_2, \ldots, m_q$ are missing, while the data at the other sampling instants $t_1, t_2, \ldots, t_p$ are available. For instance, for the irregular sampling pattern in Figure 1, the available outputs are $y(0), y(1), y(3), y(6), y(10), y(15), y(16), y(21), y(28), \ldots$, namely, $y(t_0), y(t_1), y(t_2), y(t_3), y(t_4), y(t_5), y(t_6), y(t_7), \ldots$, for $t_0 = 0, t_1 = 1, t_2 = 3, t_3 = 6, t_4 = 10, t_5 = 15, t_6 = 16, t_7 = 21, t_8 = 28, \ldots$. This is a general framework in which we assume the patterns with irregular sampling outputs availability. Specially, it includes all output availability as special cases when $t_{i+1} - t_i = 1$ for all $i$. For parameter estimation, the number of the measurable data is larger than the number of the unknown parameters, that is $p > n$.

*2.2. FIR Model: A Common Model.* FIR model is a common model, which can approximate the other kinds of models.

*2.2.1. ARX Model.* Consider the following ARX model:

$$A(z)y(t) = B(z)u(t) + v(t), \tag{3}$$

which can be simplified as follows:

$$y(t) = \frac{B(z)}{A(z)}u(t) + \frac{1}{A(z)}v(t). \tag{4}$$

Using the finite impulse response method, we can get

$$\begin{aligned} F(z) &= \frac{B(z)}{A(z)}, \\ &= f_1 z^{-1} + f_2 z^{-2} + \cdots + f_p z^{-p}. \end{aligned} \tag{5}$$

We define

$$w(t) = \frac{1}{A(z)}v(t). \tag{6}$$

The ARX model can be simplified as the following FIR model:

$$y(t) = F(z)u(t) + w(t). \tag{7}$$

*2.2.2. CARMA Model.* The CARMA model is written by

$$A(z)y(t) = B(z)u(t) + D(z)v(t), \tag{8}$$

which can be simplified as

$$y(t) = \frac{B(z)}{A(z)}u(t) + \frac{D(z)}{A(z)}v(t). \tag{9}$$

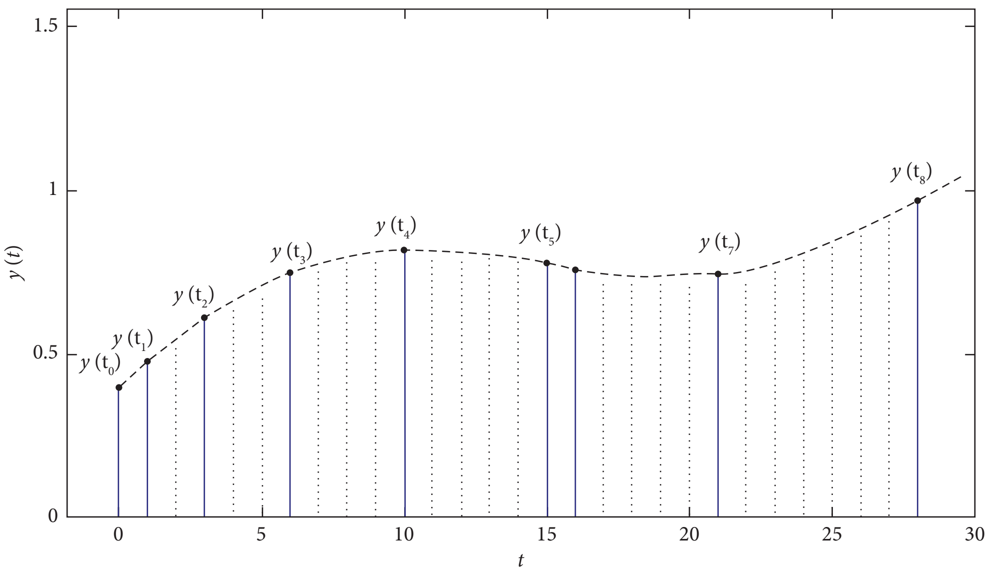
Let

$$\begin{aligned} F(z) &= \frac{B(z)}{A(z)}, \\ w(t) &= \frac{D(z)}{A(z)}v(t). \end{aligned} \tag{10}$$

Then, the CARMA model is simplified as an ARX model.

*2.2.3. OE Model.* The OE model is written by

$$y(t) = \frac{B(z)}{A(z)}u(t) + v(t). \tag{11}$$



$y(0), y(1), y(3), y(6), y(10), y(15), y(16), y(21), y(28), \cdots$ are available

Figure 1: The irregular sampling output.

Let

$$F(z) = \frac{B(z)}{A(z)},$$
$$w(t) = v(t). \tag{12}$$

Clearly, the OE model is the same as the ARX model.

*Remark 1.* Since almost all the other kinds of models can be approximated by an FIR model, we can first estimate the parameters of the FIR model and then recover parameters of the original models.

## 3. FGD-Based Auxiliary Model Algorithm

*3.1. Fractional Derivative-Review.* The traditional GD and Newton algorithms update the parameters with the first derivative and second derivative, respectively. Their corresponding convergence rates are linear convergent and quadratic convergent. Although the Newton method has faster convergence rates, it involves a matrix inverse calculation in each iteration which leads to heavy computational efforts. A nature question arises, can we use a fractional derivative to obtain an algorithm which has a faster convergence rate than the GD algorithm and has less computational efforts than the Newton method.

There are three conventional fractional derivative methods [32–34]: (1) the Riemann–Liouville (RL) fractional derivative, (2) the Caputo (CAP) fractional derivative, and (3) the Grünwald–Letnikov (GL) fractional derivative.

*3.1.1. RL Fractional Derivative Method*

$${}^{RL}_{a}D^{\alpha}_{x}f(x) \triangleq D^{n}\left({}_{a}D^{\alpha-n}_{x}f(x)\right) = \frac{1}{\Gamma(n-\alpha)}\frac{\mathrm{d}^{n}}{\mathrm{d}x^{n}}\int_{a}^{x}\frac{f(t)}{(x-t)^{\alpha-n+1}}\mathrm{d}t, \tag{13}$$

where $\Gamma(\alpha) = \alpha \times \Gamma(\alpha - 1)$.

*3.1.2. CAP Fractional Derivative Method*

$${}^{\mathrm{C}}_{a}D^{\alpha}_{x}f(x) = \frac{1}{\Gamma(n-\alpha)}\int_{a}^{x}\frac{f^{(n)}(t)}{(x-t)^{\alpha+1-n}}\mathrm{d}t. \tag{14}$$

*3.1.3. GL Fractional Derivative Method*

$$\begin{aligned}
&{}^{\mathrm{GL}}_{a}D^{\alpha}_{x}f(x)\\
&\triangleq \lim_{h\longrightarrow 0}\frac{1}{n^{\alpha}}\sum_{k=0}^{\lfloor x-a/h\rfloor}\frac{(-1)^{k}\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)}f(x-kh)\\
&= \lim_{h\longrightarrow 0}\frac{1}{n^{\alpha}}\sum_{k=0}^{\lfloor x-a/h\rfloor}\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(-\alpha)}f(x-kh).
\end{aligned} \tag{15}$$

Take $f(x) = x$, for example, the $\alpha$-order fractional derivative of $f(x)$ is written by

$$f^{\alpha}(x) = \frac{\Gamma(2)}{\Gamma(2-\alpha)}x^{1-\alpha}. \tag{16}$$

*3.2. GD Algorithm and Newton Method.* Assume that we have collected $L$ sets of data $(p + q = L)$. Define

$$\begin{aligned}
Y(L) &= [y(L), y(L-1), \ldots, y(1)]^{\mathrm{T}} \in \mathbb{R}^{L},\\
\Psi(L) &= [\psi(L), \psi(L-1), \ldots, \psi(1)] \in \mathbb{R}^{n\times L},\\
\psi(L) &= [u(L-1), \ldots, u(L-n)]^{\mathrm{T}} \in \mathbb{R}^{L},\\
V(L) &= [v(L), \ldots, v(1)]^{\mathrm{T}} \in \mathbb{R}^{L}.
\end{aligned} \tag{17}$$

Then, the FIR model is written by

$$\begin{aligned} Y(L) &= \Psi^{\mathrm{T}}(L)\vartheta + V(L), \\ \vartheta &= [f_1, f_2, \ldots, f_n]^{\mathrm{T}} \in \mathbb{R}^n. \end{aligned} \tag{18}$$

Define the cost function as

$$J(\vartheta) = \frac{1}{2}\left\| Y(L) - \Psi^{\mathrm{T}}(L)\vartheta \right\|^2. \tag{19}$$

Using the GD algorithm for the FIR model, we have

$$\begin{aligned} \vartheta_k &= \vartheta_{k-1} + \gamma\Psi(L)\left[Y(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right], \\ 0 &< \gamma < \frac{2}{\lambda_{\max}\left[\Psi(L)\Psi^{\mathrm{T}}(L)\right]}. \end{aligned} \tag{20}$$

Since the output vector $Y(L)$ contains the missing outputs $y(m_1), y(m_2), \ldots, y(m_q)$, we use the auxiliary model to compute the missing outputs, that is, the missing outputs in iteration $k$ are computed based on the parameter estimates in iteration $k-1$,

$$\hat{y}_k(m_i) = \psi^{\mathrm{T}}(m_i)\vartheta_{k-1}, i = 1, 2, \ldots, q. \tag{21}$$

Then, the GD-based auxiliary model (GD-AM) algorithm is listed as follows:

$$\begin{aligned} \vartheta_k &= \vartheta_{k-1} + \gamma\Psi(L)\left[\hat{Y}_k(L) - \Psi^{T}(L)\vartheta_{k-1}\right], \\ 0 &< \gamma < \frac{2}{\lambda_{\max}\left[\Psi(L)\Psi^{T}(L)\right]}, \\ \hat{Y}_k(L) &= [\hat{y}_k(L), \hat{y}_k(L-1), \ldots, \hat{y}_k(1)]^{T}, \\ \hat{y}_k(i) &= \begin{cases} y(i), & i = t_j, j = 1, 2, \ldots, p, \\ \hat{y}_k(i), & i = m_o, o = 1, 2, \ldots, q. \end{cases} \end{aligned} \tag{22}$$

*Remark 2.* The GD-AM algorithm should compute the eigenvalues of the information matrix to choose a suitable step-size. Two shortcomings exist: (1) if the information matrix has a high order, to compute its eigenvalues is difficult; (2) if the information matrix is ill-conditioned, whatever the step-size is, the convergence rates are quite slow.

Different from the GD-AM algorithm, the Newton-based auxiliary model (NT-AM) method updates the parameters by

$$\vartheta_k = \vartheta_{k-1} + \left[\Psi(L)\Psi^{\mathrm{T}}(L)\right]^{-1}\Psi(L)\left[\hat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right], \tag{23}$$

$$\begin{aligned} \hat{Y}_k(L) &= [\hat{y}_k(L), \hat{y}_k(L-1), \ldots, \hat{y}_k(1)]^{\mathrm{T}}, \\ \hat{y}_k(i) &= \begin{cases} y(i), & i = t_j, \\ \hat{y}_k(i), & i = m_o. \end{cases} \end{aligned} \tag{24}$$

Simplifying equation (23) yields

$$\vartheta_k = \left[\Psi(L)\Psi^{\mathrm{T}}(L)\right]^{-1}\Psi(L)\hat{Y}_k(L). \tag{25}$$

*Remark 3.* From equation (25), we can conclude that the NT-AM algorithm is the same as the least squares algorithm. Therefore, the NT-AM algorithm has faster convergence rates than the GD-AM algorithm. However, the NT-AM algorithm needs to compute the inverse of a matrix. If the matrix has a high order, to compute its inverse is challenging.

*3.3. FGD-Based Auxiliary Model Algorithm.* The FGD algorithm takes $\alpha - th$ derivative of the cost function with the aim of increasing the convergence rates and avoiding the matrix inversion. Its basic idea is to add an additional direction behind the negative gradient direction. The FGD algorithm is written by

$$\vartheta_k = \vartheta_{k-1} - r_1\frac{\partial J(\vartheta)}{\partial\vartheta} - r_2\frac{\partial^{\alpha} J(\vartheta)}{\partial\vartheta^{\alpha}}, \tag{26}$$

where

$$\frac{\partial J(\vartheta)}{\partial\vartheta} = -\Psi(L)\left[\hat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right], \tag{27}$$

is the gradient descent direction, and

$$\frac{\partial^{\alpha} J(\vartheta)}{\partial\vartheta^{\alpha}} = -\Psi(L)\left[\hat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right]\frac{\partial^{\alpha}\vartheta}{\partial\vartheta^{\alpha}}, \tag{28}$$

is the additional direction.

Based on equation (16), it follows that

$$\frac{\partial^{\alpha}\vartheta}{\partial\vartheta^{\alpha}} = \frac{1}{\Gamma(2-\alpha)}\vartheta^{1-\alpha}. \tag{29}$$

Then, the FGD algorithm is transformed into

$$\begin{aligned} \vartheta_k &= \vartheta_{k-1} + r_1\Psi(L)\left[\hat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right] \\ &\quad + r_2\Psi(L)\left[\hat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right]\frac{\partial^{\alpha}\vartheta}{\partial\vartheta^{\alpha}} \\ &= \vartheta_{k-1} + \left[r_1 + r_2\frac{1}{\Gamma(2-\alpha)}\vartheta_{t-1}^{1-\alpha}\right]\Psi(L)\left[\hat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right]. \end{aligned} \tag{30}$$

Clearly, the difference between the FGD and GD algorithms is the additional direction and its corresponding step-size. The additional direction can polish the negative gradient descent direction with the aim of obtaining a better direction.

Substituting the parameter estimates at iteration $k-1$ with the estimates at iteration $k-2$ into the second part of the right side of equation (30) yields

$$\begin{aligned} \vartheta_k &= \vartheta_{k-1} + r_1\Psi(L)\left[\hat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right] \\ &\quad + r_2\Psi(L)\left[\hat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-2}\right]\frac{\partial^{\alpha}\vartheta}{\partial\vartheta^{\alpha}}, \end{aligned} \tag{31}$$

and let

$$\begin{aligned}\gamma_1 &= r_1,\\ \gamma_2 &= r_2\frac{\partial^\alpha \vartheta}{\partial \vartheta^\alpha}.\end{aligned}\tag{32}$$

It gives rise to

$$\begin{aligned}\vartheta_k = \vartheta_{k-1} &+ \gamma_1 \Psi(L)\left[\widehat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right]\\ &+ \gamma_2 \Psi(L)\left[\widehat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-2}\right].\end{aligned}\tag{33}$$

*Remark 4.* Equation (33) is equivalent to the conjugate gradient direction (C-GD) algorithm. Therefore, the FGD algorithm can be regarded as a modified C-GD algorithm.

*3.4. Properties of the FGD Algorithm.* Subtracting the true values $\vartheta$ on both sides of equation (30) obtains

$$\begin{aligned}e_k &= e_{k-1} + r_1 \Psi(L)\left[\widehat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right]\\ &\quad + r_2 \Psi(L)\left[\widehat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right]\frac{\partial^\alpha \vartheta}{\partial \vartheta^\alpha}\\ &= e_{k-1} - \left[r_1 + r_2\frac{1}{\Gamma(2-\alpha)}\vartheta_{k-1}^{1-\alpha}\right]\Psi(L)\Psi^{\mathrm{T}}(L)e_{k-1}\\ &= \left[\mathbf{I} - \left[r_1 + r_2\frac{1}{\Gamma(2-\alpha)}\vartheta_{k-1}^{1-\alpha}\right]\Psi(L)\Psi^{\mathrm{T}}(L)\right]e_{k-1}.\end{aligned}\tag{34}$$

To keep the FGD algorithm convergent, one should guarantee that

$$\left\|\mathbf{I} - \left[r_1 + r_2\frac{1}{\Gamma(2-\alpha)}\vartheta_{k-1}^{1-\alpha}\right]\Psi(L)\Psi^{\mathrm{T}}(L)\right\| < 1. \tag{35}$$

Assume that the eigenvalues of the matrix $\Psi(L)\Psi^{\mathrm{T}}(L)$ are $0 \leqslant \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$, then we can get

$$\begin{aligned}&\mathbf{I} - \left[r_1 + r_2\frac{1}{\Gamma(2-\alpha)}\vartheta_{k-1}^{1-\alpha}\right]\Psi(L)\Psi^{\mathrm{T}}(L)\\ &\quad = P^{-1}\left[\mathbf{I} - \beta\mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}\right]P,\end{aligned}\tag{36}$$

where $P$ is a nonsingular matrix, and $\beta = r_1 + r_2(1/\Gamma(2-\alpha))\vartheta_{k-1}^{1-\alpha}$. Since

$$\left[\mathbf{I} - \beta\mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}\right] = \begin{bmatrix} 1-\beta\lambda_1 & 0 & \cdots & 0\\ 0 & 1-\beta\lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1-\beta\lambda_n\end{bmatrix}. \tag{37}$$

Therefore, to keep the FGD algorithm convergent, the step-size $\beta$ should satisfy

$$0 < \beta < \frac{2}{\lambda_n}, \tag{38}$$

and the best step-size is

$$\beta = \frac{2}{\lambda_n + \lambda_1}. \tag{39}$$

For the best step-size, the convergence factor is

$$\mu = \frac{\eta - 1}{\eta + 1}, \tag{40}$$

where $\eta$ is the condition number and is written by

$$\eta = \frac{\lambda_n}{\lambda_1}. \tag{41}$$

*Remark 5.* The same as the GD algorithm, if the condition number $\eta$ of the information matrix is large, no matter what the step-size is, the convergence rates of the FGD algorithm are quite slow.

## 4. Aitken FGD-AM Algorithm

In the FGD-AM algorithm, one should compute the eigenvalues of the matrix $\Psi(L)\Psi^{\mathrm{T}}(L)$ to obtain a suitable step-size. If the matrix has a large condition number, the convergence rate of the FGD algorithm is slow. In this section, we introduce the Aitken FGD-AM (A-FGD-AM) algorithm to increase the convergence rate of the FGD-AM algorithm.

*4.1. Aitken Acceleration Method.* Rewrite the FGD algorithm as follows:

$$\vartheta_k = \vartheta_{k-1} + \beta\Psi(L)\left[\widehat{Y}_k(L) - \Psi^{\mathrm{T}}(L)\vartheta_{k-1}\right]. \tag{42}$$

Let

$$\vartheta_k = \left[f_k^1, f_k^2, \ldots, f_k^n\right]^{\mathrm{T}}. \tag{43}$$

Using the Aitken method to obtain a new estimate sequence,

$$\overline{f}_k^i = f_k^i - \frac{\left(f_{k+1}^i - f_k^i\right)^2}{f_{k+2}^i + f_k^i - 2f_{k+1}^i}, i = 1, 2, \ldots, n. \tag{44}$$

**Theorem 1.** *Assume that the parameter estimate $\vartheta_k$ is updated by the FGD algorithm in (42), and the estimate sequence $\{\overline{\vartheta}_k\}$ is computed by the Aitken method in (44). Then, the sequence $\{\overline{\vartheta}_k\}$ is at least quadratic convergent.*

*Proof.* The FGD algorithm in equation (42) can be rewritten as

$$\vartheta_k = g(\vartheta_{k-1}). \tag{45}$$

Then, the iterative function of the Aitken method can be expressed as

$$h\left(\overline{\vartheta}_{k-1}\right) = \vartheta_{k-1} - \frac{\left(g(\vartheta_{k-1}) - \vartheta_{k-1}{}^2\right.}{g(g(\vartheta_{k-1})) + \vartheta_{k-1} - 2g(\vartheta_{k-1})}. \tag{46}$$

Clearly, the derivative of the iterative function $h(\overline{\vartheta}_{k-1})$ is

$$h^{'}\left(\overline{\vartheta}_{k-1}\right) = 0. \tag{47}$$

If

$$h^{''}\left(\overline{\vartheta}_{k-1}\right) \neq 0, \tag{48}$$

the sequence $\left\{\overline{\vartheta}_k\right\}$ is quadratic convergent; otherwise, the sequence is at least third-order convergent. This completes the proof.

The steps of the A-FGD-AM algorithm are listed as follows:

(1) To initialize, let $\vartheta_0 = \mathbf{1}/p_0$, with $\mathbf{1}$ being a column vector whose entries are all unity and $p_0 = 10^6$, and $y(t) = 0, u(t) = 0, t \leqslant 0$.
(2) Collect the input data $\{u(1), u(2), \ldots, u(L),\}$ and the measurable output data $\left\{y(o_1), y(o_2), \ldots, y(o_p)\right\}$
(3) Let $k = 1$
(4) Use the auxiliary model method to compute the missing output data $\left\{\widehat{y}_k(m_1), \widehat{y}_k(m_2), \ldots, \widehat{y}_k(m_q)\right\}$
(5) Form $\widehat{Y}_k(L)$
(6) Update the parameter estimation vector $\vartheta_k$ by (30)
(7) Use the Aitken method to obtain the parameter estimation vector $\overline{\vartheta}_k$
(8) Compare $\overline{\vartheta}_k$ and $\overline{\vartheta}_{k-1}$: if they are sufficiently close, or for some preset small $\varepsilon$, if $\|\overline{\vartheta}_k - \overline{\vartheta}_{k-1}\| \leqslant \varepsilon$, then terminate the procedure and obtain the estimate $\overline{\vartheta}_k$; otherwise, increase $k$ by 1 and go to step 4. □

*Remark 6.* By using the Aitken method, one does not require to compute the eigenvalues of the information matrix. That is, the A-FGD-AM algorithm is robust to the step-size.

*4.2. Summary.* Based on the discussion above, the properties of these three algorithms are summarized as follows:

GD-AM algorithm

*Advantage.* It has less computational efforts than the NT-AM algorithm.
*Disadvantage.* It has the slowest convergence rates and is difficult to compute the step-size.

NT-AM Algorithm

*Advantage.* It has the fastest convergence rates among these three algorithms
*Disadvantage* It needs to compute the inverse of an information matrix and has the heaviest computational efforts among these three algorithms

FGD-AM Algorithm

*Advantage.* It has faster convergence rates than the GD-AM algorithm, has less computational efforts than the NT-AM algorithm, and can use the Aitken acceleration technique to increase the convergence rates.
*Disadvantage.* It has heavier computational efforts than the GD-AM algorithm.

## 5. Examples

*5.1. Example 1.* Consider an FIR model with missing data:

$$\begin{aligned} y(t) &= f_1 u(t-1) + f_2 u(t-2) + f_3 u(t-3) \\ &\quad + f_4 u(t-4) + f_5 u(t-5) + v(t), \\ \vartheta &= \left[f_1, f_2, f_3, f_4, f_5\right]^{\mathrm{T}}, \\ &= [0.2, -0.3, 0.4, 0.3, 0.1]^{\mathrm{T}}, \\ u(t) &\sim N(0,1), v(t) \sim N\left(0, 0.1^2\right). \end{aligned} \tag{49}$$

In simulation, we collect 500 sets of input and output data, where all the input data are measurable and the output data at the sampling instants $t = 4, 8, 12, \ldots, 500$ are missing. Let $\alpha = 1.2$.

Use the GD-AM, NT-AM, and FGD-AM algorithms for this FIR model. The parameter estimates and their estimation errors $\tau = \|\vartheta_k - \vartheta\|/\|\vartheta\|$ are shown in Table 1 and Figure 2. The output estimates are shown in Figure 3.

In addition, we apply the FGD-AM algorithm with different noise to signal ratios for this FIR model (FGD-AM-1, $v(t) \sim N(0, 0.1^2)$; FGD-AM-2, $v(t) \sim N(0, 0.2^2)$; and FGD-AM-3, $v(t) \sim N(0, 0.3^2)$). The estimation errors $\tau = \|\vartheta_k - \vartheta\|/\|\vartheta\|$ are shown in Figure 4.

From this simulation, we can get the following findings:

(1) The parameter estimates of these three algorithms can asymptotically converge to the true values
(2) The NT-AM algorithm has the fastest convergence rates, then is the FGD-AM algorithm, and the GD-AM algorithm has the slowest convergence rates;
(3) The estimated outputs of these three algorithms can catch the true outputs
(4) The larger the noise to signal ratio is, the larger estimation errors the FGD-AM algorithm will have

*5.2. A Water Tank System.* Consider a simple water tank system in Figure 5, where $u(t)$ is the position of the inlet water valve, and $y(t)$ is the level of Tank 2. We try to keep the level of Tank 2 in a fixed number; if the level is larger than the fixed number, we will turn $u(t)$ down; on the other hand, we should turn $u(t)$ up. The outputs (the true level of Tank 2) are transmitted over the communication network, and some data are missing due to network congestion. Assume that the system is modeled by the following FIR model:

$$\begin{aligned} y(t) &= 0.4732u(t-1) + 0.3216u(t-2) - 0.4574u(t-3) \\ &\quad + 0.2352u(t-4) + v(t). \end{aligned} \tag{50}$$

In simulation, we use a random binary sequence generated by Matlab as the input signal $u = idinput(320,'rbs', [0,1], [-1.5, 1.5])$ and assign $\alpha = 1.4$.

Table 1: The parameter estimates and their estimation errors.

| Algorithms | $k$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $\tau(\%)$ |
|---|---|---|---|---|---|---|---|
| GD-AM | 1 | 0.01816 | −0.02790 | 0.03770 | 0.02632 | 0.01047 | 90.76400 |
| | 2 | 0.03461 | −0.05313 | 0.07189 | 0.05033 | 0.01982 | 82.38729 |
| | 5 | 0.07528 | −0.11531 | 0.15647 | 0.11050 | 0.04228 | 61.64713 |
| | 10 | 0.12134 | −0.18521 | 0.25237 | 0.18041 | 0.06621 | 38.09592 |
| | 15 | 0.14958 | −0.22761 | 0.31118 | 0.22458 | 0.07968 | 23.62824 |
| | 20 | 0.16691 | −0.25334 | 0.34724 | 0.25246 | 0.08721 | 14.74477 |
| NT-AM | 1 | 0.18741 | −0.28870 | 0.37823 | 0.29285 | 0.08355 | 5.26669 |
| | 2 | 0.19988 | −0.30138 | 0.39833 | 0.30092 | 0.09284 | 1.20668 |
| | 5 | 0.20088 | −0.30199 | 0.39963 | 0.30090 | 0.09379 | 1.06486 |
| | 10 | 0.20088 | −0.30199 | 0.39963 | 0.30090 | 0.09379 | 1.06478 |
| | 15 | 0.20088 | −0.30199 | 0.39963 | 0.30090 | 0.09379 | 1.06478 |
| | 20 | 0.20088 | −0.30199 | 0.39963 | 0.30090 | 0.09379 | 1.06478 |
| FGD-AM | 1 | 0.11235 | −0.16065 | 0.19141 | 0.13352 | 0.02828 | 51.50809 |
| | 2 | 0.16286 | −0.23592 | 0.29032 | 0.20723 | 0.05092 | 27.04707 |
| | 5 | 0.19881 | −0.29528 | 0.38311 | 0.28405 | 0.08384 | 4.59820 |
| | 10 | 0.20102 | −0.30183 | 0.39883 | 0.29989 | 0.09311 | 1.16937 |
| | 15 | 0.20089 | −0.30198 | 0.39958 | 0.30084 | 0.09375 | 1.07048 |
| | 20 | 0.20088 | −0.30199 | 0.39963 | 0.30089 | 0.09379 | 1.06513 |
| | True values | 0.20000 | −0.30000 | 0.40000 | 0.30000 | 0.10000 | |

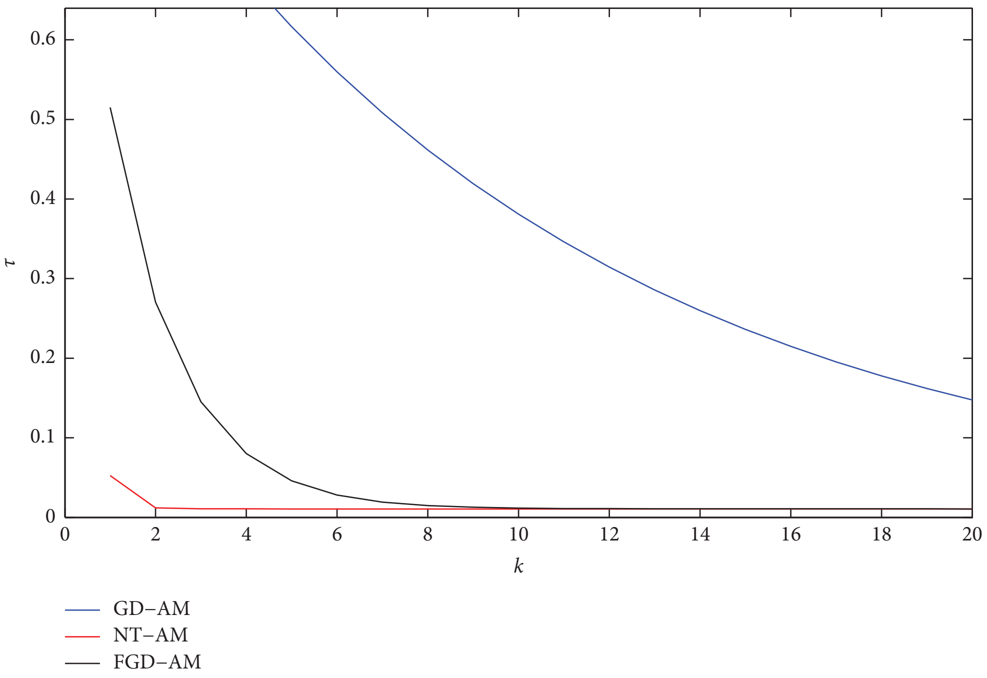


Figure 2: The parameter estimation errors.

Assume the outputs at the sampling instants $t = 3, 6, 9, \ldots, 300$ are missing, while the other outputs are measurable.

Apply the FGD-AM and A-FGD-AM algorithms with different step-sizes for this model. The parameter estimation errors $\tau = \|\vartheta_k - \vartheta\|/\|\vartheta\|$ are shown in Figures 6 and 7. The output estimates are shown in Figures 8 and 9. In addition, use the A-FGD-AM and NT-AM algorithms for the water tank system, where the A-FGD-AM algorithm chooses the optimal step-size ($\gamma = 2/\lambda_{\max}[\Psi(L)\Psi^{\mathrm{T}}(L)] + \lambda_{\min}[\Psi(L)\Psi^{\mathrm{T}}(L)]$). The parameter estimation errors are shown in Figure 10, and their elapsed times are shown in Table 2.

This simulation example shows that the A-FGD-AM algorithm has faster convergence rates than the FGD-AM algorithm; the A-FGD-AM algorithm is robust to the step-size, while the FGD-AM algorithm is sensitive to the step-size; and the A-FGD-AM algorithm has almost the same convergence rates as the NT-AM algorithm, but it has less computational efforts than the NT-AM algorithm.

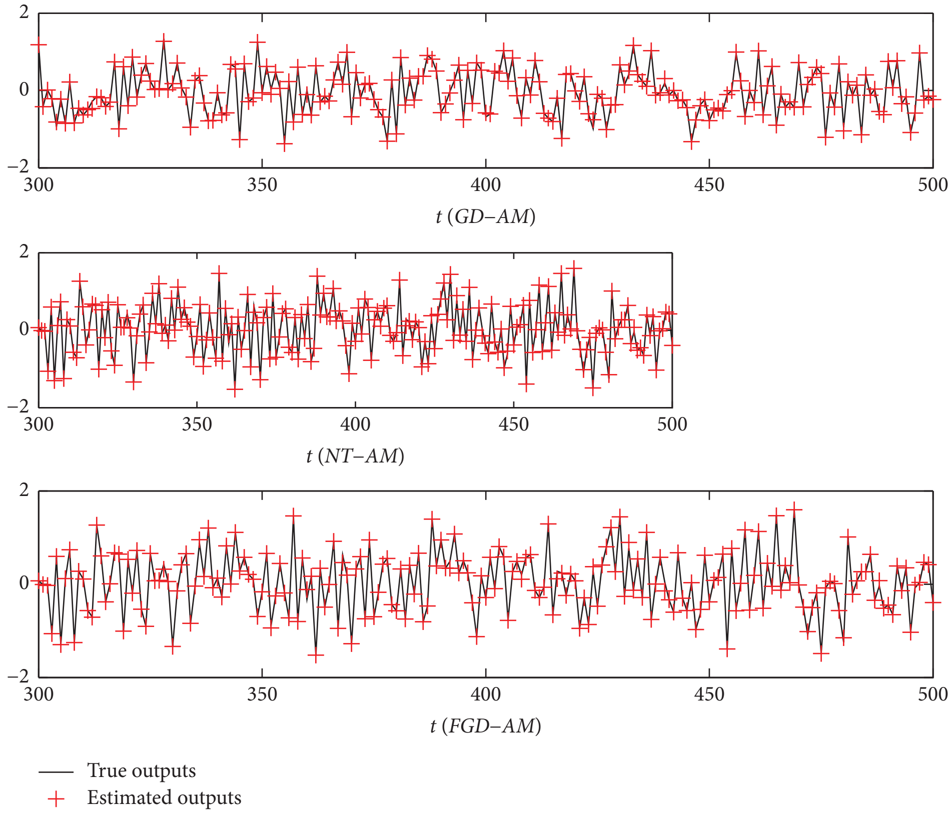


Figure 3: The true outputs and estimated outputs.

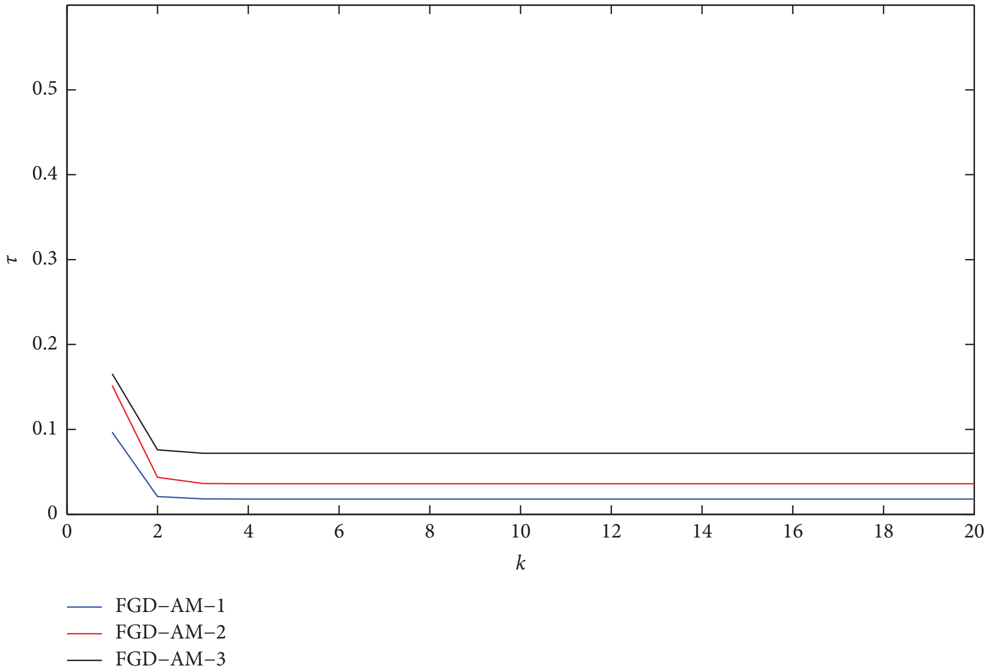


Figure 4: The parameter estimation errors with different noise-to-signal ratios.

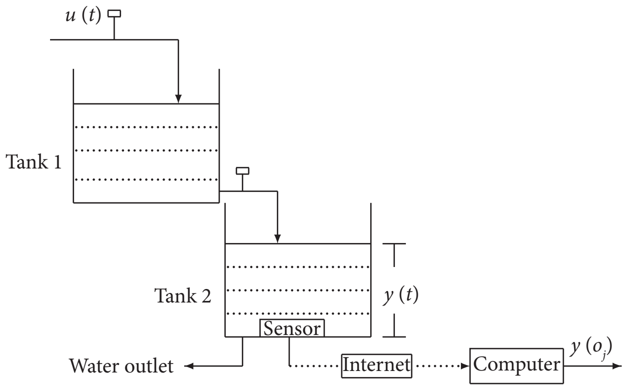


FIGURE 5: A water tank system.

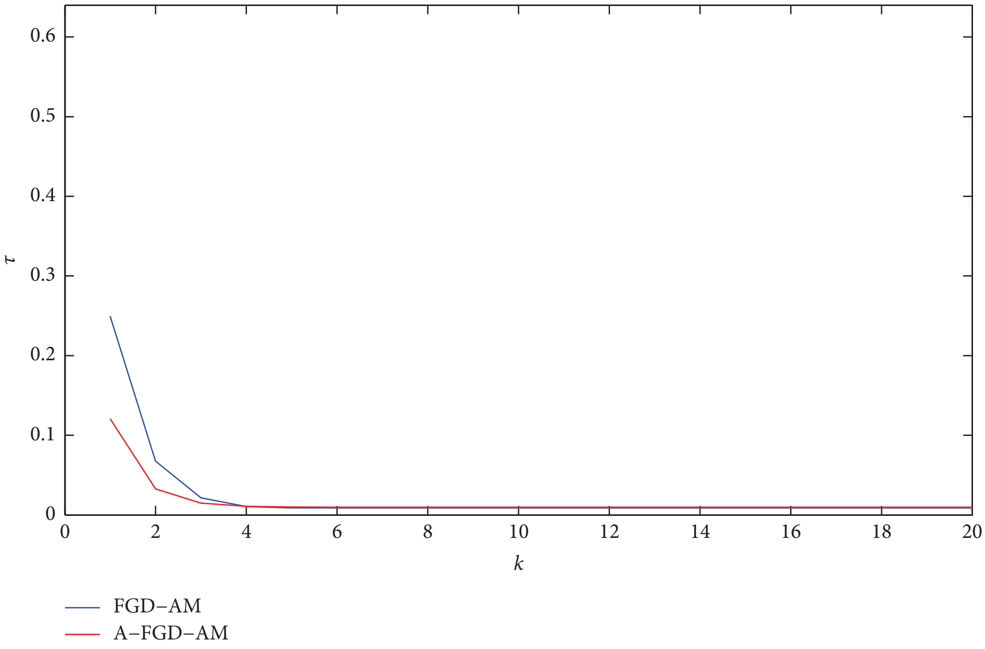


FIGURE 6: The parameter estimation errors $(\gamma = 1/\lambda_{\max}[\Psi(L)\Psi^{\mathrm{T}}(L)])$.

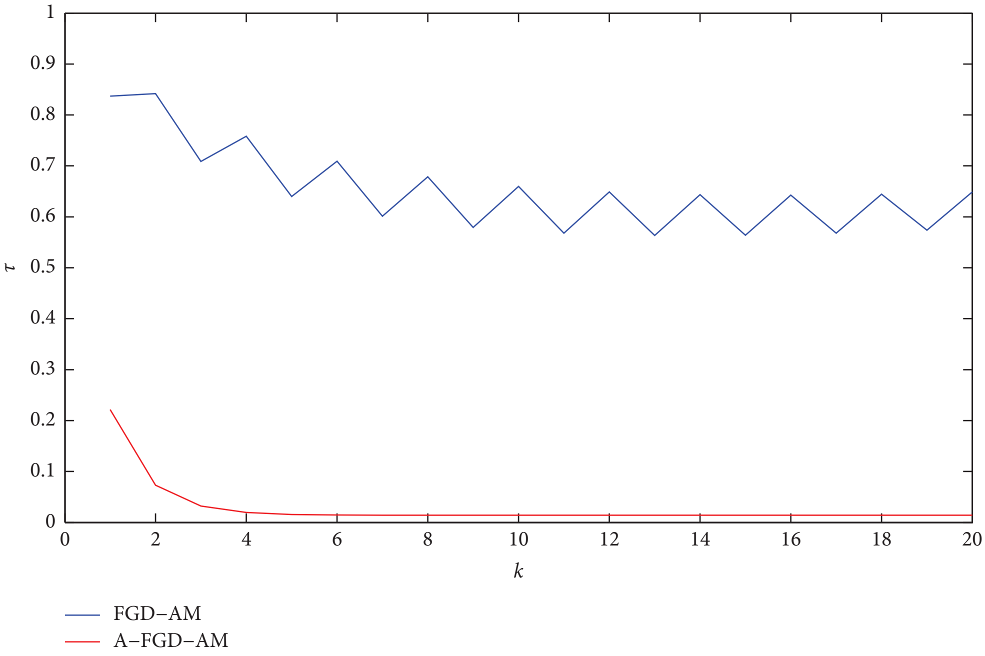


FIGURE 7: The parameter estimation errors $(\gamma = 2.01/\lambda_{\max}[\Psi(L)\Psi^{\mathrm{T}}(L)])$.

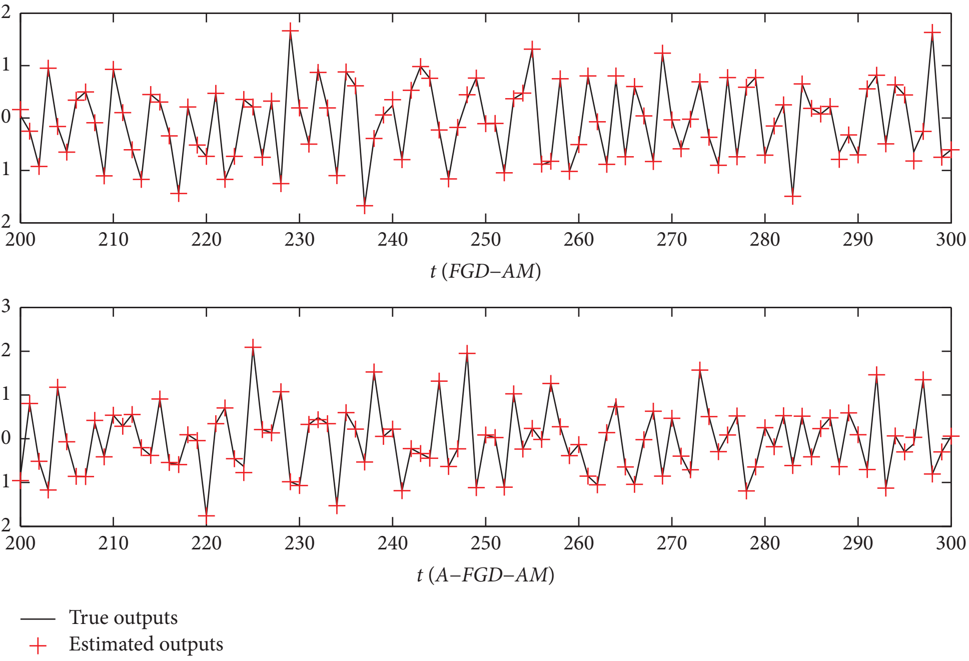


Figure 8: The true outputs and estimated outputs $(\gamma = 1/\lambda_{\max}[\Psi(L)\Psi^{T}(L)])$.

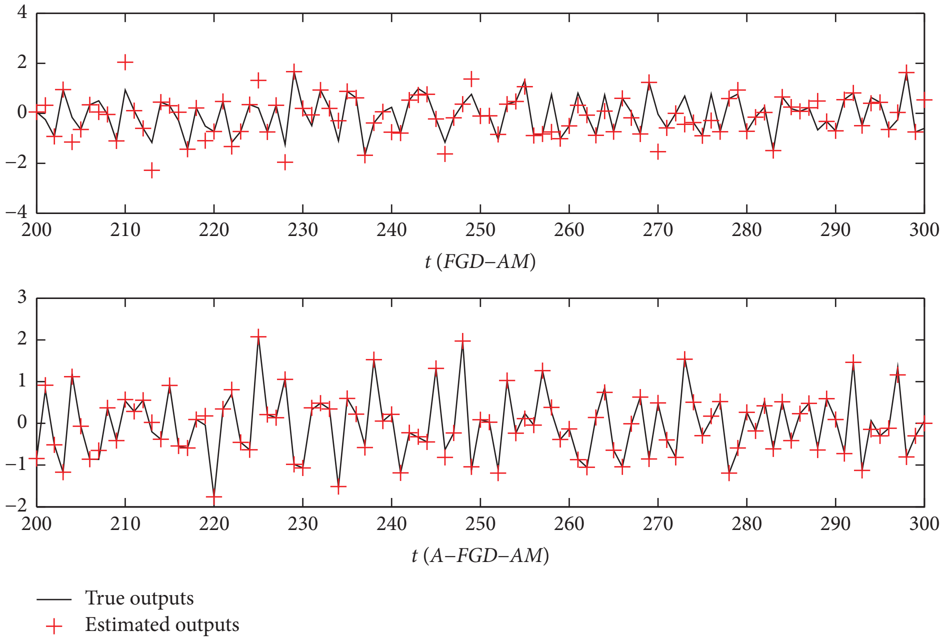


Figure 9: The true outputs and estimated outputs $(\gamma = 2.01/\lambda_{\max}[\Psi(L)\Psi^{T}(L)])$.

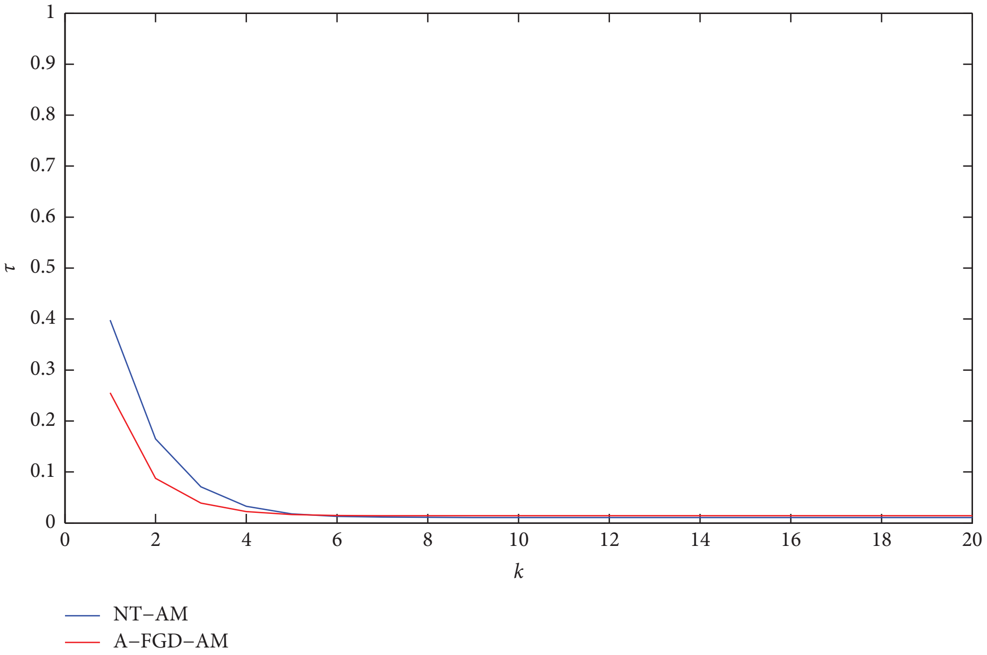


Figure 10: The parameter estimation errors.

Table 2: The elapsed times of these two algorithms.

| Algorithm | NT-AM | A-FGD-AM |
|---|---|---|
| Elapsed times (second) | 0.239 | 0.202 |

## 6. Conclusions

An FGD-AM algorithm is proposed for FIR models with missing data. By adding an additional direction to the conventional GD-AM algorithm, the convergence rates can be increased. Since the FGD-AM algorithm involves the step-size calculation in each iteration, an A-FGD-AM algorithm is developed. The A-FGD-AM algorithm has several advantages over the FGD-AM algorithm and the NT-AM algorithm, e.g., it has faster convergence rates, is robust to the step-size, and does not require matrix inverse calculation. Therefore, the A-FGD-AM algorithm can be used in a wide range of cutting-edge applications.

Although the FGD-AM algorithm has several advantages over the conventional GD-AM algorithm, some interesting topics about the FGD-AM algorithm need to be further discussed. For example, how to choose an optimal fractional order in each iteration and how to avoid the outliers in the Aitken FGD-AM algorithm.

## Data Availability

The data generated or analyzed during this study are included within the article.

## Conflicts of Interest

The author declares that they there are no conflicts of interest.

## Acknowledgments

This study was supported by the Natural Science Foundation for Colleges and Universities in Jiangsu Province (Grant no. 20KJB120032).

## References

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

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

[3] T. Soderstrom and P. Stoica, *Systen Identification*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

[4] L. Xu, F. Ding, and E. F. Yang, “Auxiliary model multi-innovation stochastic gradient parameter estimation methods for nonlinear sandwich systems,” *International Journal of Robust and Nonlinear Control*, vol. 31, no. 1, pp. 148–165, 2021.

[5] M. Jiao, D. Q. Wang, Y. Yang, and F. Liu, “More intelligent and robust estimation of battery state-of-charge with an improved regularized extreme learning machine,” *Engineering Applications of Artificial Intelligence*, vol. 104, no. 2, Article ID 104407, 2021.

[6] M. Gan, H. T. Zhu, G. Y. Chen, and C. L. P. Chen, “Weighted generalized cross validation based regularization for broad learning system,” *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 4064–4072, 2022.

[7] X. P. Liu and X. Q. Yang, “Identification of nonlinear state-space systems with skewed measurement noises,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 11, pp. 4654–4662, 2022.

[8] 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.

[9] Y. H. Zhou, X. Zhang, and F. Ding, “Hierarchical estimation approach for RBF-AR models with regression weights based

on the increasing data length," *IEEE Transactions on Circuits and Systems-II: Express Briefs*, vol. 68, no. 12, pp. 3597–3601, 2021.

[10] 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.

[11] M. Gan, X. X. Chen, F. Ding, G. Y. Chen, and C. L. P. Chen, "Adaptive RBF-AR models based on multi-innovation least squares method," *IEEE Signal Processing Letters*, vol. 26, no. 8, pp. 1182–1186, 2019.

[12] Y. Gu, Q. Zhu, J. C. Liu, P. Y. Zhu, and Y. X. Chou, "Multi-innovation stochastic gradient parameter and state estimation algorithm for dual-rate state-space systems with d-step time delay," *Complexity*, Article ID 6128697, 2020.

[13] S. Abbasbandy, A. Jafarian, and R. Ezzati, "Conjugate gradient method for fuzzy symmetric positive definite system of linear equations," *Applied Mathematics and Computation*, vol. 171, no. 2, pp. 1184–1191, 2005.

[14] Q. Tu, Y. J. Rong, and J. Chen, "Parameter identification of ARX models based on modified momentum gradient descent algorithm," *Complexity*, vol. 2020, Article ID 9537075, 11 pages, 2020.

[15] Y. M. Fan and X. M. Liu, "Data filtering-based multi-innovation forgetting gradient algorithms for input non-linear FIR-MA systems with piecewise-linear characteristics," *Journal of the Franklin Institute*, vol. 358, no. 18, pp. 9818–9840, 2021.

[16] J. Chen, J. X. Ma, M. Gan, and Q. M. Zhu, "Multi-direction gradient iterative algorithm: a unified framework for gradient iterative and least squares algorithms," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6770–6777, 2022.

[17] H. Xu, F. Ding, and E. F. Yang, "Three-stage multi-innovation parameter estimation for an exponential autoregressive time-series model with moving average noise by using the data filtering technique," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 1, pp. 166–184, 2021.

[18] N. I. Chaudhary, M. A. Z. Raja, Z. A. Khan, A. Mehmood, and S. M. Shah, "Design of fractional hierarchical gradient descent algorithm for parameter estimation of nonlinear control autoregressive systems," *Chaos, Solitons & Fractals*, vol. 157, Article ID 111913, 2022.

[19] N. I. Chaudhary, M. A. Z. Raja, Y. He, Z. A. Khan, and J. Tenreiro Machado, "Design of multi innovation fractional LMS algorithm for parameter estimation of input nonlinear control autoregressive systems," *Applied Mathematical Modelling*, vol. 93, pp. 412–425, 2021.

[20] 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.

[21] Z. A. Khan, M. A. Z. Raja, N. I. Chaudhary, K. Mehmood, and Y. He, "MISGD: moving-information-based stochastic gradient descent paradigm for personalized fuzzy recommender systems," *International Journal of Fuzzy Systems*, vol. 24, no. 1, pp. 686–712, 2022.

[22] 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, pp. 179575–179590, 2019.

[23] M. A. Z. Raja and N. I. Chaudhary, "Two-stage fractional least mean square identification algorithm for parameter estimation of CARMA systems," *Signal Processing*, vol. 107, pp. 327–339, 2015.

[24] N. I. Chaudhary, S. Zubair, and M. A. Z. Raja, "A new computing approach for power signal modeling using fractional adaptive algorithms," *ISA Transactions*, vol. 68, pp. 189–202, 2017.

[25] T. Y. Xu, J. Chen, Y. Pu, and L. X. Guo, "Fractional-based stochastic gradient algorithms for time-delayed ARX models," *Circuits, Systems, and Signal Processing*, vol. 41, no. 4, pp. 1895–1912, 2022.

[26] 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, pp. 10245–10262, 2020.

[27] 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.

[28] H. Li and Y. Shi, "Distributed receding horizon control of large-scale nonlinear systems: handling communication delays and disturbances," *Automatica*, vol. 50, no. 4, pp. 1264–1271, 2014.

[29] V. Sanjaroon, A. Farhadi, A. S. Motahari, and B. H. Khalaj, "Stabilization of nonlinear dynamic systems over limited capacity communication channels," *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3655–3662, 2020.

[30] J. Ding, L. L. Han, and X. M. Chen, "Time series AR modeling with missing observations based on the polynomial transformation," *Mathematical and Computer Modelling*, vol. 51, no. 5-6, pp. 527–536, 2010.

[31] X. Wang, F. Ding, A. Alsaedi, and T. Hayat, "Auxiliary model-based iterative parameter estimation for a nonlinear output-error system with saturation and dead-zone nonlinearity," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 9, pp. 4262–4286, 2021.

[32] R. Almeida, "A caputo fractional derivative of a function with respect to another function," *Communications in Nonlinear Science and Numerical Simulation*, vol. 44, pp. 460–481, 2017.

[33] S. S. Cheng, *Study on Fractional Order LMS Adaptive Filtering Algorithm*, University of Science and Technology of China, Anhui, China, 2018.

[34] T. T. Hartley and C. F. Lorenzo, "Fractional-order system identification based on continuous order distributions," *Signal Processing*, vol. 83, no. 11, pp. 2287–2300, 2003.