Greedy search method for separable nonlinear models using stage Aitken gradient descent and least squares algorithms

Jing Chen, Yawen Mao*, Min Gan, Dongqing Wang, Quanmin Zhu

***Abstract*—Aitken gradient descent (AGD) algorithm takes some advantages over the standard gradient descent (SGD) and Newton methods: (1) can achieve at least quadratic convergence in general; (2) does not require the Hessian matrix inversion; (3) has less computational efforts. When using the AGD method for a considered model, the iterative function should be unchanging during all the iterations. This paper proposes a hierarchical AGD algorithm for separable nonlinear models based on stage greedy method. The linear parameters are estimated using the least squares algorithm, and the nonlinear parameters are updated based on the AGD algorithm. Since the iterative function is changing at each iteration, a stage AGD algorithm is introduced. The convergence properties and simulation examples show effectiveness of the proposed algorithm.**

***Index Terms*—Parameter estimation, hierarchical identification algorithm, Aitken acceleration technique, convergence rate, separable nonlinear model**

I. Introduction

Consider a separable nonlinear model [1]–[3]

$$y(t)=f(\boldsymbol{\theta}_N,Y(t-1),U(t-1))\boldsymbol{\theta}_L+v(t), \tag{1}$$

where $y(t)$ is the output, $v(t)$ is a stochastic white noise with zero mean and variance σ^2, $Y(t-1)$ and $U(t-1)$ are the output and input data sets before the sampling instant t, respectively, $f(\cdot)$ is a nonlinear function with known structure, and $\boldsymbol{\theta}_N\in\mathbb{R}^n$ and $\boldsymbol{\theta}_L\in\mathbb{R}^m$ are nonlinear and linear parameter vectors, respectively. Assume that we have collected S input and output data $\{u(1),y(1),\cdots,u(S),y(S)\}$ $(S>m+n)$. Define

$$\begin{aligned}Y(S)&=[y(S),y(S-1),\cdots,y(1)]^{\mathrm{T}},\\ F(\boldsymbol{\theta}_N,S)&=[f(\boldsymbol{\theta}_N,Y(S-1),U(S-1)),f(\boldsymbol{\theta}_N,Y(S-2),\\ &\qquad U(S-2)),\cdots,f(\boldsymbol{\theta}_N,Y(0),U(0))]^{\mathrm{T}},\\ V(S)&=[v(S),v(S-1),\cdots,v(1)]^{\mathrm{T}}.\end{aligned}$$

Then, it gives rise to

$$Y(S)=F(\boldsymbol{\theta}_N,S)\boldsymbol{\theta}_L+V(S).$$

The focus of this paper is to use the generated input and output data $\{u(1),y(1),\cdots,u(S),y(S)\}$ to estimate the parameters $\boldsymbol{\theta}=\{\boldsymbol{\theta}_N,\boldsymbol{\theta}_L\}$.

If a derivative equation of a cost function has analytical solutions, the least squares (LS) algorithm can obtain the optimal estimates in only one iteration [4]–[6]. However, the majority of nonlinear equations are not solvable analytically. Using the LS algorithm for such models can be problematic [7]. Fortunately, the gradient descent (GD) algorithm is a great choice which does not require solving a derivative equation, thus it can be applied to nonlinear models with complex structures [8]–[10]. Designing a suitable step-size plays an important role when running a GD algorithm: a small step-size usually leads to slow convergence rates, while a larger one may cause divergence. In [11], an optimal step-size is given which involves the eigenvalue calculation. However, calculating the eigenvalues is challenging for high-order information matrices.

In the separable nonlinear model, the nonlinear parameters are embedded in the structures [12]–[14]. Therefore, the LS algorithm cannot simultaneously estimate the nonlinear and linear parameters. The GD algorithm can be efficient but with quite slow convergence rates. To increase the convergence rates, the Newton, quasi-Newton and Gauss-Newton (GN) methods are applied to these nonlinear models, these methods have faster convergence rates with the cost of heavier computational costs [15]–[17]. Recently, researchers usually use the hierarchical technique to decompose the separable nonlinear model into two submodels: one is a linear model, and the other is a nonlinear model, and then separately identify these two submodels [18], [19]. For example, in [20], the linear parameters are estimated using the LS algorithm, and the nonlinear parameters are estimated based on the GD algorithm. Different from the work in [20], Gan proposed a variable projection GN (VP-GN) algorithm for a separable nonlinear model, and by which the parameter estimates can quickly converge to the true values [12]. Since the VP method may lead to complex nonlinear cost function and the GN method involves the matrix inversion, the VP-GN method can be inefficient for systems who have high-order nonlinear parameter vectors or complex nonlinear structures [3].

The Aitken method is an efficient technique which can increase the convergence rates from linear convergence to at least quadratic convergence [21], [22]. Its basic idea is to increase the convergence rates using dual-drive technology: one is the original iterative function generated by a linear convergent algorithm, e.g., the GD method [23], the power method [24]; and the other is a transformed function which is constructed by the original iterative function. Compared with the Newton method, the Aitken based method avoids the matrix inverse calculation, thus can be extended to nonlinear systems with high-order. However, it has an assumption that the original function should be unchanged during the iterations [25]. Once the systems have hidden variables, the Aitken accelerating method would be inefficient. For example, in the hierarchical identification algorithms, the linear parameter estimates are changing at each iteration, using the Aitken method to increase the convergence rate is unavailable [26], [27].

In this paper, a greedy search method using stage AGD and LS algorithms is proposed for separable nonlinear models. An LS algorithm and a stage AGD algorithm are interactively used to estimate the linear and nonlinear parameters. At each iteration, we aim to obtain the optimal estimates under current data. For instance, the LS algorithm can obtain the optimal linear parameter estimates when the

J. Chen and Y.W. Mao are with School of Science, Jiangnan University, Wuxi 214122, PR China (8201703038@jiangnan.edu.cn, myw0530@163.com).

M. Gan is with College of Computer Science and Technology, Qingdao University, Qingdao 266071, PR China (aganmin@aliyun.com).

D.Q. Wang is with College of Electrical Engineering, Qingdao University, Qingdao 266071, PR China (dqwang64@163.com).

Q.M. Zhu is with Department of Engineering Design and Mathematics, Bristol BS16 1QY, UK (quan.zhu@uwe.ac.uk).

This work is supported by the National Natural Science Foundation of China (Nos. 61973137, 62073082), the Natural Science Foundation of Jiangsu Province (No. BK20201339) and the Taishan Scholar Program of Shandong Province (No. tsqn201909095).

nonlinear parameters are fixed, and the stage AGD algorithm can get the better nonlinear parameter estimates when compared with the GD algorithm. For this reason, this algorithm is termed as greedy search method, and it sets the following aims (1) has fast convergence rates but with less computational efforts; (2) does not require the matrix inverse calculation; (3) avoids calculating the eigenvalues.

Briefly, this paper is organized as follows. Section II reviews the traditional algorithms. Section III develops the hierarchical stage AGD (H-AGD) algorithm. The properties of the H-AGD algorithm are given in Section IV. Section V provides the simulation examples. Finally, Section VI sums up the paper and gives future directions.

II. Review–Traditional algorithms

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

Rewrite the separable nonlinear model as

$$Y(S) = F(\boldsymbol{\theta}_N, S)\boldsymbol{\theta}_L + V(S). \tag{2}$$

Then, several traditional identification algorithms which can estimate the parameters $\boldsymbol{\theta}_N$ and $\boldsymbol{\theta}_L$ are introduced.

A. Joint algorithm

Assume that the parameters at iteration $k-1$ are $\boldsymbol{\theta}_N^{k-1}$ and $\boldsymbol{\theta}_L^{k-1}$, next, we want to get the estimates at iteration k.

In the joint algorithm, the linear and nonlinear parameters are simultaneously estimated. Define the cost function

$$J(\boldsymbol{\theta}_L, \boldsymbol{\theta}_N) = \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N, S)\boldsymbol{\theta}_L\|^2.$$

Since the nonlinear parameters are embedded in the nonlinear function, the derivative equation of the above cost function does not have analytical solutions. The joint-LS algorithm is difficult for this separable nonlinear model. The widely used methods are the joint-GD and joint-GN methods [28]–[30].

1. Joint-GD algorithm

To update the parameters at iteration k, the negative gradient direction is computed as

$$d_k = -\left[\begin{array}{c} J'(\boldsymbol{\theta}_L^{k-1}) \\ J'(\boldsymbol{\theta}_N^{k-1}) \end{array}\right] \in \mathbb{R}^{m+n}.$$

Then, the parameter estimates can be updated by

$$\left[\begin{array}{c} \boldsymbol{\theta}_L^{k} \\ \boldsymbol{\theta}_N^{k} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{\theta}_L^{k-1} \\ \boldsymbol{\theta}_N^{k-1} \end{array}\right] + \gamma_k d_k,$$

where γ_k is the step-size.

Remark 1: Actually, to find a suitable step-size is challenging, especially for the system with high-order/complex nonlinear structure. For example, if $Y(S) = F(S)\boldsymbol{\theta}, \boldsymbol{\theta} \in \mathbb{R}^{m+n}$, one should compute the eigenvalues of a $(m+n)$-order matrix; while in [28], for a complex nonlinear model, the authors use a small step-size to avoid calculating the eigenvalues for finding a suitable step-size.

2. Joint-GN algorithm

The GD algorithm has slow convergence rates for its zigzagging nature. To increase the convergence rates, the GN algorithm is a good choice.

Using the GN algorithm for the nonlinear model yields

$$\left[\begin{array}{c} \boldsymbol{\theta}_L^{k} \\ \boldsymbol{\theta}_N^{k} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{\theta}_L^{k-1} \\ \boldsymbol{\theta}_N^{k-1} \end{array}\right] + H_k^{-1} d_k, \tag{3}$$

where the Hessian matrix is

$$H_k = \left[\begin{array}{cc} J''(\boldsymbol{\theta}_L^{k-1}) & \{J'(\boldsymbol{\theta}_L^{k-1})\}^{\mathrm{T}} J'(\boldsymbol{\theta}_N^{k-1}) \\ \{J'(\boldsymbol{\theta}_L^{k-1})\}^{\mathrm{T}} J'(\boldsymbol{\theta}_N^{k-1}) & J''(\boldsymbol{\theta}_N^{k-1}) \end{array}\right] \in \mathbb{R}^{(m+n)\times(m+n)}.$$

Remark 2: In the joint-GN algorithm, one should perform a matrix inverse calculation at each iteration. If the matrix has a high-order, to compute its inverse is difficult [3].

B. Hierarchical identification algorithm

The key of the hierarchical identification algorithm is first to separate the complex nonlinear model into two sub-models: one is a linear-parameter-model and the other is a nonlinear-parameter-model, and then to update the linear and nonlinear parameters interactively [31], [32].

Assume that the nonlinear parameter estimates at iteration $k-1$ are $\boldsymbol{\theta}_N^{k-1}$. Define the cost function

$$J_L(\boldsymbol{\theta}_L) = \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N^{k-1}, S)\boldsymbol{\theta}_L\|^2.$$

Using the LS algorithm to get the linear parameter estimates yields

$$\boldsymbol{\theta}_L^k = [F^{\mathrm{T}}(\boldsymbol{\theta}_N^{k-1}, S)F(\boldsymbol{\theta}_N^{k-1}, S)]^{-1}F^{\mathrm{T}}(\boldsymbol{\theta}_N^{k-1}, S)Y(S).$$

Then, the nonlinear parameters $\boldsymbol{\theta}_N^k$ will be estimated based on $\boldsymbol{\theta}_L^k$.

1. Hierarchical-GD (H-GD) algorithm for estimating $\boldsymbol{\theta}_N^k$

Define the cost function

$$J_N(\boldsymbol{\theta}_N) = \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N, S)\boldsymbol{\theta}_L^k\|^2.$$

The negative gradient direction is

$$d_k = [F(\boldsymbol{\theta}_N^{k-1}, S)\boldsymbol{\theta}_L^k]'[Y(S) - F(\boldsymbol{\theta}_N^{k-1}, S)\boldsymbol{\theta}_L^k].$$

It follows that

$$\boldsymbol{\theta}_N^k = \boldsymbol{\theta}_N^{k-1} + \gamma_k d_k,$$

where γ_k is the step-size which can be determined based on an n-order matrix.

Remark 3: Compared with the joint-GD algorithm, computing the step-size of the H-GD algorithm is easier because the order of the parameter vector is reduced from $m+n$ to n.

Remark 4: In this paper, we assume that the nonlinear cost function $J_N(\boldsymbol{\theta}_N) = \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N, S)\boldsymbol{\theta}_L^k\|^2$ is strictly convex for a fixed $\boldsymbol{\theta}_L^k$, which means that the cost function has only one stable point.

2. Hierarchical-GN (H-GN) algorithm for estimating $\boldsymbol{\theta}_N^k$

Let

$$J_N'(\boldsymbol{\theta}_N) = \frac{\partial J_N(\boldsymbol{\theta}_N)}{\partial \boldsymbol{\theta}_N},\ J_N''(\boldsymbol{\theta}_N) = \frac{\partial^2 J_N(\boldsymbol{\theta}_N)}{\partial \boldsymbol{\theta}_N^2}.$$

The parameter estimates using the GN method can be written by

$$\boldsymbol{\theta}_N^k = \boldsymbol{\theta}_N^{k-1} - [J_N''(\boldsymbol{\theta}_N^{k-1})]^{-1} J_N'(\boldsymbol{\theta}_N^{k-1}),$$

where $J_N''(\boldsymbol{\theta}_N^{k-1}) \in \mathbb{R}^{n\times n}$ is a Hessian matrix.

Remark 5: In the H-GN method, the order of the Hessian matrix is n which is smaller than $m+n$. Therefore, it is more efficient than the joint-GN algorithm.

C. VP algorithm

The VP algorithm uses a function of $\boldsymbol{\theta}_N$ to express the linear parameters $\boldsymbol{\theta}_L$, and then substitutes the function into the original system which only contains the nonlinear parameters [33], [34].

First, based on the LS algorithm, the linear parameters are expressed by

$$\boldsymbol{\theta}_L = [F^{\mathrm{T}}(\boldsymbol{\theta}_N, S)F(\boldsymbol{\theta}_N, S)]^{-1}F^{\mathrm{T}}(\boldsymbol{\theta}_N, S)Y(S). \tag{4}$$

Substituting the above equation into (2) yields

$$\begin{aligned} Y(S) = F(\boldsymbol{\theta}_N, S)[F^{\mathrm{T}}(\boldsymbol{\theta}_N, S)F(\boldsymbol{\theta}_N, S)]^{-1}F^{\mathrm{T}}(\boldsymbol{\theta}_N, S)Y(S) + \\ V(S). \end{aligned} \tag{5}$$

Clearly, the above model only contains the nonlinear parameters $\boldsymbol{\theta}_N$. Define the following cost function

$$\begin{aligned} J(\boldsymbol{\theta}_N) = \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N, S) \times \\ [F^{\mathrm{T}}(\boldsymbol{\theta}_N, S)F(\boldsymbol{\theta}_N, S)]^{-1}F^{\mathrm{T}}(\boldsymbol{\theta}_N, S)Y(S)\|^2. \end{aligned} \tag{6}$$

Then, utilize the GD/GN algorithm to update the nonlinear parameters $\boldsymbol{\theta}_N$. Once, an optimal nonlinear parameter vector estimate is obtained, the linear parameter vector can be yielded according to Equation (4).

Remark 6: Unlike the hierarchical identification algorithm, the VP algorithm first gets the nonlinear parameter estimates using an iterative function, then obtains the linear parameter estimates based on the nonlinear parameter estimates in only one iteration.

Remark 7: The VP algorithm can reduce the order of the model from $m+n$ to n. However, such a reduced order nonlinear model has a more complex structure, that is, to compute the step-size or the Hessian matrix inversion is more challenging when compared with the hierarchical identification algorithm.

D. Summary

From the above subsections, the properties of these three kinds of algorithms are listed as follows:

(1) Joint algorithm

Advantages: (1) can simultaneously estimate the linear and nonlinear parameters; (2) has faster convergence rates than its corresponding partners in the hierarchical identification algorithm;

Disadvantages: (1) has the heaviest computational efforts among these three kinds of algorithms; (2) needs to compute the eigenvalues or the inverse of a high-order matrix $(m+n)$.

(2) Hierarchical identification algorithm

Advantages: (1) has the simplest iterative function among these three kinds of algorithms; (2) has less computational efforts than its corresponding partners in the joint algorithm;

Disadvantages: (1) has the slowest convergence rates among these three kinds of algorithms; (2) needs to compute the eigenvalues or the inverse of a low-order matrix (n).

(3) VP algorithm

Advantages: (1) has faster convergence rates than its corresponding partners in the hierarchical identification algorithm; (2) has less computational efforts than its corresponding partners in the joint algorithm;

Disadvantages: (1) has the most complex iterative functions among these three kinds of algorithms; (2) needs to compute the eigenvalues or the inverse of a low-order matrix (n).

III. Greedy search method

Since the hierarchical identification algorithm has the simplest structure but the slowest convergence rates. In this section, we propose a hierarchical AGD (H-AGD) algorithm, which is based on greedy search method. This algorithm, combing the Aitken method, can overcome the shortcomings of the H-GD and H-GN methods.

A. Aitken method

Define an iterative function as

$$x_k = f(x_{k-1}).$$

Assume that the sequence $\{x_k\}$ generated using the above iterative function is linear convergent. The basic idea of the Aitken method is to obtain a new sequence $\{\bar{x}_k\}$ based on the original sequence $\{x_k\}$. Such a new sequence can be computed by

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

The above equation is equivalent to

$$\bar{x}_k = x_k - \frac{(f(x_k)-x_k)^2}{f(f(x_k))+x_k-2f(x_k)}.$$

Lemma 1: For a convergent sequence $\{x_k\}$ generated by an iterative function $f(x)$, if the first derivative of function $f(x)$ satisfies

$$0<|f'(x)|<1,$$

the sequence $\{x_k\}$ is linear convergent; and when $|f'(x)|=0$, the sequence $\{x_k\}$ is at least quadratic convergent.

Proof: Assume that the true value is x_*, since the sequence $\{x_k\}$ is convergent, we have

$$x_* = f(x_*).$$

Subtracting x_* on both sides of $x_k = f(x_{k-1})$ yields

$$e_k = f'(\epsilon)e_{k-1},\ e_k = x_k - x_*,$$

where ϵ lays between x_{k-1} and x_*. It follows that

$$0<\frac{|e_k|}{|e_{k-1}|} = |f'(\epsilon)|<1,$$

which means that the sequence is linear convergent. In addition,

$$e_k = f(x_{k-1})-f(x_*) = f'(x_*)(x_{k-1}-x_*)+\frac{f''(\epsilon)}{2}(x_{k-1}-x_*)^2,$$

when $|f'(x)|=0$, it gives rise to

$$\frac{|e_k|}{|e^2(k-1)|} = \frac{1}{2}|f''(\epsilon)|.$$

$f''(\epsilon)\neq 0$, the sequence is quadratic convergent. That is, the sequence $\{x_k\}$ is at least quadratic convergent when $|f'(x)|=0$. ■

Based on Lemma 1, we can get the following theorem.

Theorem 1: For an iterative function defined as

$$g(x) = x - \frac{(f(x)-x)^2}{f(f(x))+x-2f(x)},$$

if its first derivative exists, and $\lim_{x\to x_*} f'(x)\neq 1$. Then, the sequence $\{x_k\}$ generated by the iterative function $x_k = g(x_{k-1})$ is at least quadratic convergent.

Proof: When $x\to x_*$, we have

$$f(x_*) = x_*,\ f(f(x_*)) = 2f(x_*)-x_*.$$

The first derivative of function $g(x)$ is

$$\lim_{x\to x_*} g'(x) = 1 - \lim_{x\to x_*}\frac{\alpha(x)}{\beta(x)}, \tag{7}$$

where $\alpha(x)$ and $\beta(x)$ are written by

$$\begin{aligned}\alpha(x) = {} & 2(f(x)-x)(f'(x)-1)(f(f(x))+x-2f(x)) - \\ & (f(x)-x)^2(f'(f(x))f'(x)+1-2f'(x)),\\ \beta(x) = {} & (f(f(x))+x-2f(x))^2.\end{aligned}$$

Equation (7) is simplified as

$$\lim_{x\to x_*} g'(x) = 1 - \lim_{x\to x_*}\frac{0}{0}. \tag{8}$$

Using L'hospital's rule for the second part of the right side of Equation (8) yields

$$\lim_{x\to x_*}\frac{\alpha(x)}{\beta(x)} = \lim_{x\to x_*}\frac{(f'(x)-1)^2}{(f'(x)-1)^2}.$$

Since $\lim_{x\to x_*} f'(x)\neq 1$, it follows that

$$\lim_{x\to x_*} g'(x) = 0.$$

This shows that the sequence $\{x_k\}$ is at least quadratic convergent. ■

Remark 8: When using the Aitken method to increase the convergence rates, the original iterative function $f(x)$ should be unchanging during all the iterations; otherwise, the Aitken method will be inefficient.

B. Aitken GD algorithm

For a fixed linear parameter vector $\boldsymbol{\theta}_L$, the nonlinear parameters updated using the GD algorithm can be written as

$$\boldsymbol{\theta}_N^k = \boldsymbol{\theta}_N^{k-1} + \gamma_k [F(\boldsymbol{\theta}_N^{k-1}, S)\boldsymbol{\theta}_L]'[Y(S) - F(\boldsymbol{\theta}_N^{k-1}, S)\boldsymbol{\theta}_L]. \quad (9)$$

Rewrite the parameter estimates as

$$\boldsymbol{\theta}_N^k = [\theta_N^k(1), \theta_N^k(2), \cdots, \theta_N^k(n)]^{\mathrm{T}},$$

while the estimates using the Aitken method are written by

$$\bar{\boldsymbol{\theta}}_N^k = [\bar{\theta}_N^k(1), \bar{\theta}_N^k(2), \cdots, \bar{\theta}_N^k(n)]^{\mathrm{T}}.$$

Each element in $\bar{\boldsymbol{\theta}}_N^k$ is computed as

$$\bar{\theta}_N^k(i) = \theta_N^k(i) - \frac{(\theta_N^{k+1}(i) - \theta_N^k(i))^2}{\theta_N^{k+2}(i) + \theta_N^k(i) - 2\theta_N^{k+1}(i)}, \quad i = 1, 2, \cdots, n. \quad (10)$$

Lemma 2: For a fixed linear parameter vector $\boldsymbol{\theta}_L$, the parameter estimates $\boldsymbol{\theta}_N^k$ using the traditional GD algorithm are computed by (9), while the parameter estimates $\bar{\boldsymbol{\theta}}_N^k$ using the Aitken method are written by (10). Then, the sequence $\bar{\boldsymbol{\theta}}_N^k$ is at least quadratic convergent.

(The proof of Lemma 2 is straightforward and hence omitted.)

Lemma 2 shows that the sequence $\{\bar{\boldsymbol{\theta}}_N^k\}$ is at least quadratic convergent if the linear parameter vector $\boldsymbol{\theta}_L$ keeps unchanging. However, in the separable nonlinear model identification, the linear parameter vector $\boldsymbol{\theta}_L$ is updated using the LS algorithm at each iteration. Thus, using the Aitken method to increase the convergence rates is invalid.

C. Aitken method using stage greedy search technique

The Aitken method can (1) increase the convergence rates from linear convergence to at least quadratic convergence; (2) be robust to the step-size, that is, whatever the step-size is, the algorithm is always convergent [25]. In order to take full advantage of the Aitken method, we introduce a stage greedy search technique for the Aitken method.

Assume that the parameter estimates at iteration $k-1$ are $\boldsymbol{\theta}_L^{k-1}$ and $\boldsymbol{\theta}_N^{k-1}$. The linear and nonlinear parameter vectors are then updated as follows:

(1) *L-step*

Using the LS algorithm to update the linear parameter vector yields

$$\boldsymbol{\theta}_L^k = [F^{\mathrm{T}}(\boldsymbol{\theta}_N^{k-1}, S)F(\boldsymbol{\theta}_N^{k-1}, S)]^{-1}F^{\mathrm{T}}(\boldsymbol{\theta}_N^{k-1}, S)Y(S).$$

(2) *N-step*

Let

$$\boldsymbol{\theta}_N^{k,0} = \boldsymbol{\theta}_N^{k-1},$$

where the index 0 means the initial parameter estimates at stage k.

Apply the GD algorithm to estimate the nonlinear parameter vector

$$\boldsymbol{\theta}_N^{k,l} = \boldsymbol{\theta}_N^{k,l-1} + \gamma_k [F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'[Y(S) - F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k], \quad l = 1, 2, \cdots, M, \quad (11)$$

$$\boldsymbol{\theta}_N^{k,l} = [\theta_N^{k,l}(1), \theta_N^{k,l}(2), \cdots, \theta_N^{k,l}(n)]^{\mathrm{T}}.$$

Equation (11) shows that the nonlinear parameter estimates are updated M iterations at stage k, and during the stage k, the iterative function keeps unchanging. Then, the Aitken method is introduced to obtain the better estimates, that is

$$\bar{\theta}_N^{k,l}(i) = \theta_N^{k,l}(i) - \frac{(\theta_N^{k,l+1}(i) - \theta_N^{k,l}(i))^2}{\theta_N^{k,l+2}(i) + \theta_N^{k,l}(i) - 2\theta_N^{k,l+1}(i)}, \quad l = 0, 1, \cdots, M-2, \ i = 1, 2, \cdots, n,$$

$$\bar{\boldsymbol{\theta}}_N^{k,l} = [\bar{\theta}_N^{k,l}(1), \bar{\theta}_N^{k,l}(2), \cdots, \bar{\theta}_N^{k,l}(n)]^{\mathrm{T}}.$$

Remark 9: Based on the Aitken method, the parameter vector estimate $\bar{\boldsymbol{\theta}}_N^{k,M-2}$ is more accurate than the estimate $\boldsymbol{\theta}_N^{k,1}$ using the GD algorithm. In addition, it does not require performing matrix inversion, thus, it has less computational efforts when compared to the GN method.

Remark 10: In the *L-step*, we update the linear parameter estimates $\boldsymbol{\theta}_L^k$ using the LS algorithm, which are the optimal estimates under current input-output data and the fixed nonlinear estimates $\boldsymbol{\theta}_N^{k-1}$; and in the *N-step*, we first use the GD algorithm to obtain several estimates of $\boldsymbol{\theta}_N$ at this stage, and then try to utilize the Aitken method to find the optimal nonlinear parameter estimates $\bar{\boldsymbol{\theta}}_N^{k,M-2}$ under current input-output data and the estimated linear parameters $\boldsymbol{\theta}_L^k$. Both the linear and nonlinear estimates are the optimal estimates under current data, this means *'greedy search'*. Therefore, the proposed H-AGD algorithm is based on stage greedy search method.

The H-AGD algorithm consists of the following iterations:

1) Let $u(t) = 0, y(t) = 0, v(t) = 0,\ t \leqslant 0$, and give a small positive number ε.
2) Let $r_0 = 1$, $k = 1$, $\boldsymbol{\theta}_N^0 = \mathbf{1}/p_0$ with $\mathbf{1}$ being a column vector whose entries are all unity and $p_0 = 10^6$.
3) Collect S input and output data $u(1), u(2), \cdots, u(S), y(1), y(2), \cdots, y(S)$.

Greedy search method using LS and AGD algorithms

4) Update the linear parameter estimates $\boldsymbol{\theta}_L^k$ using LS algorithm.
5) Let $\boldsymbol{\theta}_N^{k,0} = \boldsymbol{\theta}_N^{k-1}$.
6) Estimate the nonlinear parameter estimates $\boldsymbol{\theta}_N^{k,l}$ $l = 1, 2, \cdots, M$ based on GD algorithm.
7) Use the Aitken method to obtain $\bar{\boldsymbol{\theta}}_N^{k,l}$, $l = 0, 1, \cdots, M-2$.
8) Let $\boldsymbol{\theta}_N^k = \bar{\boldsymbol{\theta}}_N^{k,M-2}$.

9) Let $\boldsymbol{\theta}^k = [\boldsymbol{\theta}_L^k; \boldsymbol{\theta}_N^k]$.
10) Compare $\boldsymbol{\theta}^k$ with $\boldsymbol{\theta}^{k-1}$, if $\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1}\|/\|\boldsymbol{\theta}^k\| \leqslant \varepsilon$, then terminate the procedure and obtain $\boldsymbol{\theta}^k$; otherwise, increase k by 1 and go to step 4).

Remark 11: For simplicity, we do not need to compute $\bar{\boldsymbol{\theta}}_N^{k,l}$, $l = 0, 1, \cdots, M-3$ because the estimates using the Aitken method are independent on each other. That is, we can only calculate $\bar{\boldsymbol{\theta}}_N^{k,M-2}$ at stage k.

Remark 12: The H-AGD algorithm tries to obtain the optimal linear and nonlinear parameter estimates at each iteration, thus it has faster convergence rates than the H-GD algorithm. However, to choose an optimal M in the *N-step* is a challenging problem: a small M may lead to slow convergence rates, while a large one can lead to heavy computational efforts.

IV. Properties of the H-AGD algorithm

In this section, some properties of the H-AGD algorithm are provided to help readers for their ad hoc research and applications.

A. Convergence analysis

Define the cost function as

$$J(\boldsymbol{\theta}_L, \boldsymbol{\theta}_N) = \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N, S)\boldsymbol{\theta}_L\|^2.$$

Then, we can get the following theorem.

Theorem 2: Assume that the parameter estimates at iteration $k-1$ are $\boldsymbol{\theta}_L^{k-1}$ and $\boldsymbol{\theta}_N^{k-1}$, and the linear parameter estimates $\boldsymbol{\theta}_L^k$ at iteration k are updated using the LS algorithm, the nonlinear parameter estimates $\boldsymbol{\theta}_N^k$ are estimated based on the stage AGD algorithm. Then, the following inequality holds

$$J(\boldsymbol{\theta}_L^k, \boldsymbol{\theta}_N^k) \leqslant J(\boldsymbol{\theta}_L^{k-1}, \boldsymbol{\theta}_N^{k-1}).$$

Proof: The cost function at iteration $k-1$ is

$$J(\boldsymbol{\theta}_L^{k-1}, \boldsymbol{\theta}_N^{k-1}) = \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N^{k-1}, S)\boldsymbol{\theta}_L^{k-1}\|^2.$$

Fixing the nonlinear estimate $\boldsymbol{\theta}_N^{k-1}$ obtains the following cost function

$$J(\boldsymbol{\theta}_L, \boldsymbol{\theta}_N^{k-1}) = \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N^{k-1}, S)\boldsymbol{\theta}_L\|^2.$$

The LS algorithm can ensure

$$\boldsymbol{\theta}_L^k = \arg\min_{\boldsymbol{\theta}_L} \left\{ \frac{1}{2}\|Y(S) - F(\boldsymbol{\theta}_N^{k-1}, S)\boldsymbol{\theta}_L\|^2 \right\},$$

which means that

$$J(\boldsymbol{\theta}_L^k, \boldsymbol{\theta}_N^{k-1}) \leqslant J(\boldsymbol{\theta}_L^{k-1}, \boldsymbol{\theta}_N^{k-1}).$$

Once $\boldsymbol{\theta}_L^k$ is obtained, the AGD algorithm can guarantee that

$$J(\boldsymbol{\theta}_L^k, \boldsymbol{\theta}_N^k) \leqslant J(\boldsymbol{\theta}_L^k, \boldsymbol{\theta}_N^{k-1}).$$

Therefore, we have

$$J(\boldsymbol{\theta}_L^k, \boldsymbol{\theta}_N^k) \leqslant J(\boldsymbol{\theta}_L^{k-1}, \boldsymbol{\theta}_N^{k-1}).$$

■

Remark 13: Since the cost function $J(\boldsymbol{\theta}_L^k, \boldsymbol{\theta}_N^k)$ is monotonically decreasing, if it has only one stable point, Theorem 2 can ensure the parameter estimate sequence $\{\boldsymbol{\theta}_L^k, \boldsymbol{\theta}_N^k\}$ to converge to the true values.

B. Step-size choosing method

In the H-GD method, one should carefully choose a suitable step-size γ_k to keep the algorithm converging.

Rewrite the H-GD algorithm as follows,

$$\boldsymbol{\theta}_N^{k,l} = \boldsymbol{\theta}_N^{k,l-1} + \gamma_k [F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'[Y(S) - F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]. \tag{12}$$

Subtracting the true value $\boldsymbol{\theta}_N$ on both sides of the above equation yields

$$\begin{aligned} \mathbf{e}_N^{k,l} = &\mathbf{e}_N^{k,l-1} + \gamma_k [F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]' \times \\ &[F(\boldsymbol{\theta}_N, S)\boldsymbol{\theta}_L^k - F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k + V(S)]. \end{aligned}$$

Since $V(S)$ is a Gaussian white noise, for a large S, it gives rise to

$$\begin{aligned} \mathbf{e}_N^{k,l} = &\mathbf{e}_N^{k,l-1} + \gamma_k [F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]' \times \\ &[F(\boldsymbol{\theta}_N, S)\boldsymbol{\theta}_L^k - F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]. \end{aligned}$$

Using the Taylor series to simplify the above equation yields

$$\mathbf{e}_N^{k,l} \approx \mathbf{e}_N^{k,l-1} - \gamma_k [F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'\{[F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'\}^{\mathrm{T}}\mathbf{e}_N^{k,l-1}. \tag{13}$$

Lemma 3: For a linear parameter vector $\boldsymbol{\theta}_L^k$, the parameter estimates $\boldsymbol{\theta}_N^{k,l}$ using the H-GD algorithm are computed by (12). When the step-size γ_k satisfies

$$0 < \gamma_k < \frac{2}{\lambda_{max}[[F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'\{[F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'\}^{\mathrm{T}}]},$$

the H-GD algorithm is linear convergent.

Proof: According to Equation (13), the estimation errors can be written by

$$\mathbf{e}_N^{k,l} = [\mathbf{I} - \gamma_k [F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'\{[F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'\}^{\mathrm{T}}]\mathbf{e}_N^{k,l-1}.$$

When

$$0 < \gamma_k < \frac{2}{\lambda_{max}[[F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'\{[F(\boldsymbol{\theta}_N^{k,l-1}, S)\boldsymbol{\theta}_L^k]'\}^{\mathrm{T}}]}.$$

The estimation errors are simplified as

$$\|\mathbf{e}_N^{k,l}\| = \rho\|\mathbf{e}_N^{k,l-1}\|,$$

where $0 < \rho < 1$. It shows that the H-GD algorithm is linear convergent. ■

Remark 14: When using the H-GD algorithm, the step-size should be calculated at each iteration to keep the algorithm convergent. This will lead to heavy computational efforts. However, according to the work in [25], the H-AGD method has no limitation on the step-size.

Remark 15: At each stage k, one can use the Aitken method to obtain the best estimates $\bar{\boldsymbol{\theta}}_N^{k,M-2}$ under current linear parameter estimates $\boldsymbol{\theta}_L^k$ if M is large enough. That is why the method is termed as greedy search method. However, the 'best' estimates generally cannot lead to better linear parameter estimates. For example, in the steepest GD algorithm, the best step-size is computed for a negative gradient direction, but the convergence rates are quite slow because of the 'greedy' property. Therefore, we usually assign a suitable number for M.

The procedure of the H-AGD algorithm is shown in Fig. 1.

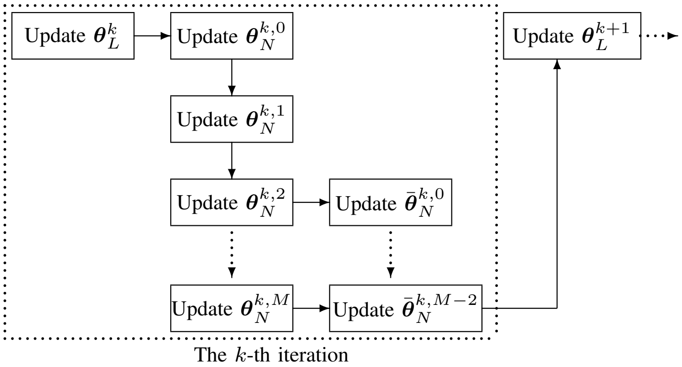
Update $\boldsymbol{\theta}_L^k$
Update $\boldsymbol{\theta}_N^{k,0}$
Update $\boldsymbol{\theta}_N^{k,1}$
Update $\boldsymbol{\theta}_N^{k,2}$
Update $\bar{\boldsymbol{\theta}}_N^{k,0}$
Update $\boldsymbol{\theta}_N^{k,M}$
Update $\bar{\boldsymbol{\theta}}_N^{k,M-2}$
Update $\boldsymbol{\theta}_L^{k+1}$
The k-th iteration

Fig. 1. The procedure of the H-AGD algorithm

C. Ill-conditioned parameter estimates in Aitken method

In the stage AGD procedure, the better parameter estimates are written by

$$\begin{aligned} \bar{\theta}_N^{k,l}(i) = \theta_N^{k,l}(i) - \frac{(\theta_N^{k,l+1}(i) - \theta_N^{k,l}(i))^2}{\theta_N^{k,l+2}(i) + \theta_N^{k,l}(i) - 2\theta_N^{k,l+1}(i)}, \\ l = 0, 1, \cdots, M-2, \ i = 1, 2, \cdots, n. \end{aligned} \tag{14}$$

Owing to the truncation error of the computer or some special cases in the identification procedure, the term in the denominator of the above equation sometimes is so close to zero, but the term in the numerator is not. For example, the true value $\theta_N(i) = 0.8$, using the GD algorithm yields $\theta_N^{k,l+2}(i) = 0.6$, $\theta_N^{k,l+1}(i) = 0.5$, and $\theta_N^{k,l}(i) = 0.4$. In this case, the sequence $\{\theta_N^{k,l}(i), l = 1, 2, \cdots\}$ generated by the GD algorithm is convergent, but the term in the denominator of (14) equals to zero, while the term in the numerator equals to 0.01. According to Equation (14), the 'better' estimate is ill-conditioned, see Figs. 3 and 8 in [25].

Since the better estimates $\bar{\theta}_N^{k,l}(i), l = 0, 1, \cdots$ are independent on each other. If $\bar{\theta}_N^{k,l_1}(i)$ is ill-conditioned, its neighboring estimate $\bar{\theta}_N^{k,l_1+1}(i)$ is usually well-conditioned. For example, in Fig. 3 of [25], the 'better' estimates at iterations 26 and 53 are ill-conditioned, but their neighboring estimates at iterations 27 and 54 quickly become well-conditioned. In general, if the iteration M is large enough, the few ill-conditioned points can be neglected. However, in this paper, M is given in prior, and the AGD algorithm is used to update the nonlinear parameters at each stage k. The ill-conditioned estimates easily exist. If $\bar{\boldsymbol{\theta}}_N^{k,M-2}$ is ill-conditioned, the next linear parameter estimates $\bar{\boldsymbol{\theta}}_L^{k+1}$ at stage $k+1$ are correspondingly ill-conditioned, which leads to divergence of the H-AGD algorithm; see Table III in Section V. To deal with this problem, two ways are introduced:

(1) For a fixed iteration number M

Compare the last two neighboring cost functions $J(\bar{\boldsymbol{\theta}}_N^{k,M-2}, \boldsymbol{\theta}_L^k)$ and $J(\bar{\boldsymbol{\theta}}_N^{k,M-3}, \boldsymbol{\theta}_L^k)$ at stage k, if

$$J(\bar{\boldsymbol{\theta}}_N^{k,M-2}, \boldsymbol{\theta}_L^k) > J(\bar{\boldsymbol{\theta}}_N^{k,M-3}, \boldsymbol{\theta}_L^k),$$

let

$$\bar{\boldsymbol{\theta}}_N^{k,M-2} = \bar{\boldsymbol{\theta}}_N^{k,M-3}.$$

(2) For a varying iteration number M

Compare the two neighboring cost functions, and if

$$J(\bar{\boldsymbol{\theta}}_N^{k,M-2}, \boldsymbol{\theta}_L^k) > J(\bar{\boldsymbol{\theta}}_N^{k,M-3}, \boldsymbol{\theta}_L^k).$$

Let $M = M+1$, and compute the new $\bar{\boldsymbol{\theta}}_N^{k,M-2}$.

Remark 16: Different from Remark 11, we should compute at least two better estimates $\bar{\boldsymbol{\theta}}_N^{k,M-2}$ and $\bar{\boldsymbol{\theta}}_N^{k,M-3}$ at stage k to avoid the ill-conditioned parameter estimates in Aitken method.

V. Examples

A. Example 1

Consider a complex exponential model [3],

$$\begin{aligned} y(t) &= b_1 e^{-a_2 u^2(t-1)} \cos(a_3 u(t-1)) + \\ &\quad b_2 e^{-a_1 u^2(t-1)} \cos(a_2 u(t-2)) + \\ &\quad b_3 e^{-a_4 u^2(t-1)} \sin(a_1 u(t-3)) + v(t), \\ \boldsymbol{\theta}_L &= [b_1, b_2, b_3]^{\mathrm{T}} = [2, 3, 2]^{\mathrm{T}}, \\ \boldsymbol{\theta}_N &= [a_1, a_2, a_3, a_4]^{\mathrm{T}} = [1, 1.5, 3, 0.8]^{\mathrm{T}}, \end{aligned}$$

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

In simulation, 1000 sets data are collected. Apply the H-GD ($\gamma_k = \frac{1}{\lambda_{max}}$), H-GN and H-AGD ($\gamma_k = 0.001$, $M = 8$) algorithms to the proposed model. For fair comparison, the initial parameters $\boldsymbol{\theta}^0 = \mathbf{1}/10^6$ keep unchanging for all the algorithms. The estimation errors $\tau := \|\boldsymbol{\theta}^k - \boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ versus k are shown in Fig. 2. The parameter estimates and the estimation errors are shown in Table I. The boxplot of parameter estimates of different iterations are shown in Fig. 3. The elapsed times of these three algorithms are shown in Table II (**by Intel(R) Core(TM) i5-7220U: 2.50GHz, 2.71GHz; RAM: 8.0 GB; Windows 10**).

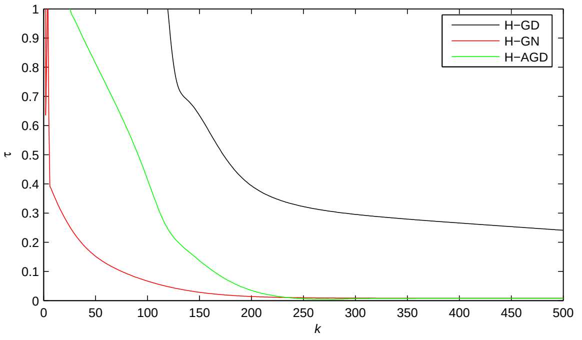

Fig. 2. The parameter estimation errors τ versus k

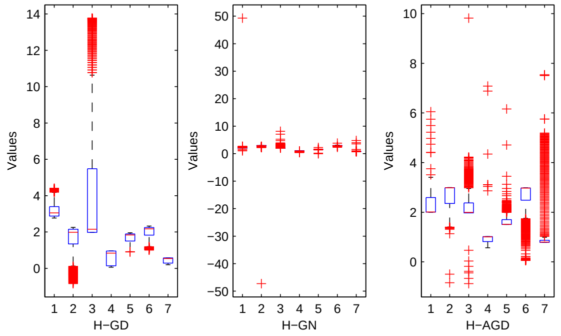

Fig. 3. The parameter estimates using different algorithms for 500 iterations

TABLE II
The Elapsed Times

Algorithm	H-GD	H-GN	H-AGD
Time (second)	4.652	7.255	10.099

Furthermore, use the H-AGD ($M = 6$) algorithm for this model, the H-AGD-1 algorithm does not compare the neighboring cost functions but the H-AGD-2 algorithm does. The parameter estimation errors using these two algorithms are shown in Table III.

The following findings can be obtained based on this simulation example:

1) The estimates using the three algorithms can asymptotically converge to the true values with an increased number of iteration k, and among them, the H-GN algorithm has the fastest convergence rates, see Fig. 2 and Table I. However, the H-GN algorithm involves the matrix inversion at each iteration.
2) When the iteration $k > 250$, the H-AGD and H-GN algorithms have almost the same accurate parameter estimates, while the H-GD algorithm has the poorest estimation accuracy, see Table I and Fig. 3.
3) Although the H-AGD algorithm has the largest elapsed time in this example, see Table II. However, as the order n increases, it will have the smallest elapsed time for the reason that there is no eigenvalue/matrix inverse calculation at each iteration.
4) The H-AGD algorithm usually has ill-conditioned nonlinear estimates, which can cause divergence of the H-AGD algorithm. We can use the method in Section IV-C to deal with this problem, see Table III.

B. Example 2: the Canadian lynx data

In this example, we consider the following separable nonlinear model which is usually applied to describe the lynx population [35],

$$\begin{aligned} y(t) &= b_0 + b_1 y(t-1) + b_2 y(t-2) + b_3 e^{a_1 (y(t-2)-3.6259)^2} + \\ &\quad b_4 e^{a_1 (y(t-2)-3.6259)^2} y(t-1) + \\ &\quad b_5 e^{a_1 (y(t-2)-3.6259)^2} y(t-2) + v(t), \\ \boldsymbol{\theta} &= [b_0, b_1, b_2, b_3, b_4, b_5, a_1]^{\mathrm{T}} \\ &= [0.4584, 1.2433, -0.3491, 0.3059, 0.3693, -0.5790, -6.0978]^{\mathrm{T}}. \end{aligned}$$

First, apply the H-GD, H-GN and H-AGD algorithms for the lynx population model, the parameter estimation errors $\tau := \|\boldsymbol{\theta}^k - \boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ are shown in Fig. 4. The boxplot of parameter estimates of different iterations are shown in Fig. 5.

Furthermore, use the joint-GD and joint-GN algorithms for the model, the parameter estimation errors are shown in Fig. 6.

This example shows that the H-GN method has the fastest convergence rates among the hierarchical identification algorithms (H-GD, H-GN and H-AGD), see Figs. 4 and 5; However, the H-GN algorithm should perform the matrix inverse calculation at each iteration. The H-AGD algorithm has faster convergence rates, and it does not require

TABLE I
THE PARAMETER ESTIMATES AND THEIR ESTIMATION ERRORS

	k	b_1	b_2	b_3	a_1	a_2	a_3	a_4	τ (%)
H-GD	50	3.96530	-0.15811	13.75762	0.06805	0.98064	1.26840	0.23440	228.84515
	100	3.59635	0.53366	10.62024	0.07753	1.14076	1.51385	0.26093	170.03619
	200	3.23980	1.82138	2.72847	0.61199	1.95058	2.15952	0.46155	39.36372
	400	2.85576	2.17531	1.98131	0.95997	1.83479	2.26599	0.57653	26.60171
	500	2.77310	2.25451	1.98018	0.96159	1.79246	2.32931	0.58081	24.11852
H-GN	50	2.46021	2.52852	2.26534	0.84900	1.64171	2.62689	0.73574	15.20153
	100	2.20554	2.79892	2.00419	0.97567	1.56299	2.80639	0.70606	6.68483
	200	2.03176	2.96510	1.97716	1.01128	1.51469	2.95546	0.77086	1.40778
	400	2.00719	2.98828	1.97401	1.01675	1.50847	2.97713	0.78204	0.83073
	500	2.00696	2.98850	1.97398	1.01680	1.50841	2.97733	0.78215	0.82686
H-AGD	50	3.03094	1.66953	3.60395	0.66384	2.12323	1.17009	4.03699	81.25330
	100	2.66388	2.10423	3.17891	0.68164	1.81161	2.03509	1.97510	41.48197
	200	2.09525	2.89017	2.05575	0.97084	1.53238	2.89893	0.80016	3.48761
	400	2.00455	2.99072	1.97410	1.01714	1.50779	2.97964	0.78351	0.78001
	500	2.00571	2.98965	1.97404	1.01697	1.50809	2.97853	0.78286	0.80190
	True Values	2.00000	3.00000	2.00000	1.00000	1.50000	3.00000	0.80000	

TABLE III
THE PARAMETER ESTIMATES AND THEIR ESTIMATION ERRORS

	k	b_1	b_2	b_3	a_1	a_2	a_3	a_4	τ (%)
H-AGD-1	50	0.02154	4.14399	1.46719	2.05956	0.00003	-26.59226	25.33907	705.27034
	100	0.06418	3.11210	2.36137	1.85976	-0.00007	-26.25757	4.25835	541.00687
	200	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	400	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	500	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
H-AGD-2	50	3.32143	1.69602	2.66486	0.88791	2.18190	0.61988	11.60533	205.95531
	100	2.99424	1.96352	2.63880	0.92437	1.97001	1.03863	10.70629	187.14010
	200	2.47533	2.36001	2.73525	0.93638	1.74312	1.43223	8.66423	148.08160
	400	1.69772	3.01092	3.06165	0.78532	1.46971	2.65077	3.53416	54.45433
	500	1.94192	3.05285	1.95053	1.03843	1.49236	3.03559	0.80014	1.95428
	True Values	2.00000	3.00000	2.00000	1.00000	1.50000	3.00000	0.80000	

the eigenvalue and matrix inverse calculations. Therefore, the H-AGD algorithm is the most efficient algorithm among the three hierarchical identification algorithms if n is large.

Fig. 6 shows that the joint-GN method can quickly obtain the parameter estimates, but it should perform a 7-order matrix inversion at each iteration, while in the H-GN method, we only calculate the inverse of a 1-order matrix at each iteration.

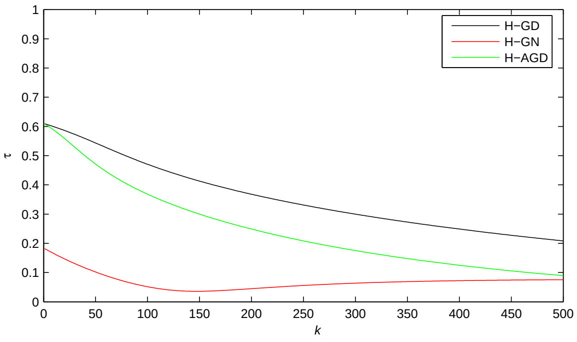

Fig. 4. The parameter estimation errors τ versus k

VI. Conclusions

In this paper, we propose a stage greedy search method for separable nonlinear models, where the linear parameters are updated using the LS algorithm, and the nonlinear parameters are estimated based on the AGD algorithm. Both these two kinds of algorithms aim to obtain the optimal estimates under current data. Compared with the traditional identification algorithms, this algorithm has the following advantages:

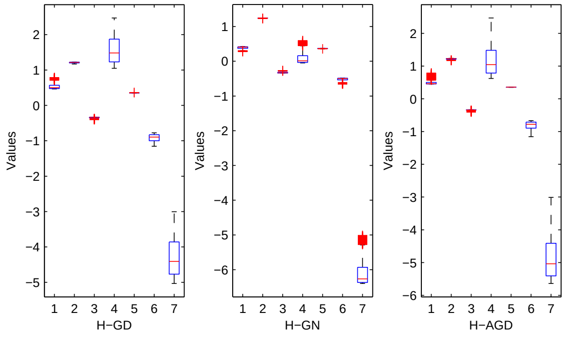

Fig. 5. The parameter estimates using different algorithms for 500 iterations

1) It has faster convergence rates than the traditional hierarchical identification algorithm, and has a simpler structure than the VP algorithm.
2) It does not require the eigenvalue calculation, then can be applied to systems with high-order.
3) It does not need to calculate the inverse of a Hessian matrix,

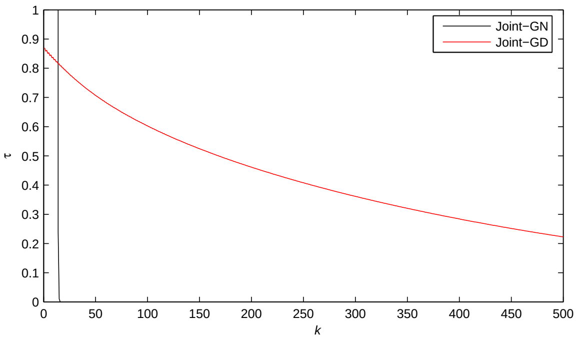

Fig. 6. The parameter estimation errors τ versus k

thus has less computational efforts than the joint-GN and H-GN algorithms.

Therefore, the proposed algorithm will have positive impact to control theories and applications as well.

Although the proposed algorithm has several advantages over the traditional algorithms, there are still some challenging and interesting topics need to be further discussed. For example, how to choose the optimal iteration number M in the AGD procedure? and can the algorithm converge to the global optimal point when the linear and nonlinear parameters are intensively coupled? These issues remain as open problems.

Acknowledgements

The authors would like to express their gratitude to the editors and anonymous reviewers for their helpful comments and constructive suggestions regarding the revision of this paper.

References

[1] J.M. Li and F. Ding, "Fitting nonlinear signal models using the increasing-data criterion," *IEEE Signal Process. Lett.*, vol. 29, pp. 1302-1306, 2022.

[2] L. Xu, "Separable multi-innovation Newton iterative modeling algorithm for multi-frequency signals based on the sliding measurement window," *Circuits Syst. Signal Process.*, vol. 41, no. 2, pp. 805-830, 2022.

[3] M. Gan, Y. Guan, G.Y. Chen, and C.L.P. Chen, "Recursive variable projection algorithm for a class of separable nonlinear models," *IEEE Trans. Neur. Net. Lear. Syst.*, vol. 32, no. 12, pp. 4971-4982, 2021.

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

[5] T. Söderström and P. Stoica, *Systen Identification*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[6] T.S. Chen, S.A. Martin, et al., "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 2933-2945, 2014.

[7] X.P. Liu and X.Q. Yang, "Identification of nonlinear state-space systems with skewed measurement noises," *IEEE Trans. Circuits Syst. I.*, 2022. DOI: 10.1109/TCSI.2022.3193444.

[8] C.P. Yu, J. Chen, and M. Verhaegen, "Subspace identification of individual systems in a large-scale heterogeneous network," *Automatica*, vol. 109, 2019. DOI: 10.1016/j.automatica.2019.108517

[9] J. Chen, Q.M. Zhu, et al., "Interval error correction auxiliary model based gradient iterative algorithms for multi-rate ARX models," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4385-4392, 2020.

[10] J. Chen, Q.M. Zhu, and Y.J. Liu, "Modified Kalman filtering based multi-step-length gradient iterative algorithm for ARX models with random missing outputs," *Automatica*, vol. 118, 2020. DOI: 10.1016/j.automatica.2020.109034

[11] J. Chen, M. Gan, Q.M. Zhu, and Y.W. Mao, "Varying infimum gradient descent algorithm for agent-server systems with uncertain communication network," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021. DOI: 10.1109/TIM.2021.3070602

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

[13] H. Xu, F. Ding, and B. Champagne, "Joint parameter and time-delay estimation for a class of nonlinear time-series models," *IEEE Signal Process. Lett.*, vol. 29, pp. 947-951, 2022.

[14] Y.H. Zhou and X. Zhang, "Partially-coupled nonlinear parameter optimization algorithm for a class of multivariate hybrid models," *Appl. Math. Comput.*, vol. 414, 2022. DOI: 10.1016/j.amc.2021.126663

[15] D.Q. Wang, L.W. Li, Y. Ji, and Y.R. Yan, "Model recovery for Hammerstein systems using the auxiliary model based orthogonal matching pursuit method," *Appl. Math. Model.*, vol. 54, pp. 537-550, 2018.

[16] X.Y. Cao and L.F. Lai, "Distributed approximate Newton's method robust to byzantine attackers," *IEEE Trans. Signal Process.*, vol. 68, pp. 6011-6025, 2020.

[17] T. Nonomura, S. Ono, K. Nakai, and Y. Saito, "Randomized subspace Newton convex method applied to data-driven sensor selection problem," *IEEE Signal Process. Lett.*, vol. 28, pp. 284-288, 2021.

[18] T. Okatani and K. Deguchi, "On the wiberg algorithm for matrix factorization in the presence of missing components," *Int. J. Comput. Vision*, vol. 72, no. 3, pp. 329-337, 2007.

[19] L. Xu, F. Ding, and Q.M. Zhu, Separable synchronous multi-innovation gradient-based iterative signal modeling from on-line measurements," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022. DOI: 10.1109/TIM.2022.3154797

[20] H. Xu, F. Ding, et al., "Two-stage recursive identification algorithms for a class of nonlinear time series models with colored noise," *Int. J. Robust Nonlinear Control*, vol. 30, no. 17, pp. 7766-7782, 2020.

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

[22] O. Bumbariu, "A new Aitken type method for accelerating iterative sequences," *Appl. Math. Comput.*, vol. 219, pp. 78-82, 2012.

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

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

[25] J. Chen, Q.M. Zhu, et al., "Robust standard gradient descent algorithm for ARX models using Aitken acceleration technique," *IEEE Trans. Cybern.*, 2021. DOI: 10.1109/TCYB.2021.3063113

[26] T. Okatani, T. Yoshida, and K. Deguchi, "Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms," *2011 Int. Conf. Comput. Vision*, Nov. 6-13, pp. 842-849, 2011.

[27] D.Q. Wang, Q.H. Fan, and Y. Ma, "An interactive maximum likelihood estimation method for multivariable Hammerstein systems," *J. Frankl. Inst.*, vol. 357, no. 17, pp. 12986-13005, 2020.

[28] M. Gan, H. Peng, et al., "A locally linear RBF network-based state-dependent AR model for nonlinear time series modeling," *Inform. Sciences*, vol. 180, pp. 4370-4383, 2010.

[29] L.S. Ngia and J. Sjoberg, "Efficient training of neural nets for nonlinear adaptive filtering using a recursive levenberg-marquardt algorithm," *IEEE Trans. Signal Process.*, vol. 48, no. 7, pp. 1915-1927, 2000.

[30] S.S. Shamsudin and X. Chen, "Recursive gauss-newton based training algorithm for neural network modelling of an unmanned rotorcraft dynamics," *Int. J. Intell. Syst. Tech. Appl.*, vol. 13, no. 1-2, pp. 56-80, 2014.

[31] F. Ding and T. Chen, "Hierarchical gradient-based identification of multivariable discrete-time systems," *Automatica*, vol. 41, no. 2, pp. 315-325, 2005.

[32] Y.J. 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.

[33] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," *Inverse Probl.*, vol. 19, no. 2, pp. R1-R26, 2003.

[34] H. Peng, T. Ozaki, V. Haggan-Ozaki, and Y. Toyoda, "A parameter optimization method for radial basis function type models," *IEEE Trans. Neural Networ.*, vol. 14, no. 2, pp. 432-438, 2003.

[35] M. Gan, C.L.P. Chen, L. Chen, and C.Y. Zhang, "Exploiting the interpretability and forecasting ability of the RBF-AR model for nonlinear time series," *Int. J. Syst. Sci.*, vol. 47, no. 8, pp. 1868-1876, 2016.