Gradient iterative algorithm for rational models based on Gram-Schmidt orthogonalization method

Yan Pu\textsuperscript{a,b}, Jing Chen\textsuperscript{a}, Yongqing Yang\textsuperscript{a}, Quanmin Zhu\textsuperscript{c}

\textsuperscript{a}School of Science, Jiangnan University, Wuxi 214122, PR China
\textsuperscript{b}School of Internet of Things Engineering, Jiangnan University, Wuxi, 214122, PR China
\textsuperscript{c}Department of Engineering Design and Mathematics, University of the West of England, Bristol BS16 1QY, UK

Abstract

An improved gradient iterative algorithm, termed as Gram-Schmidt orthogonalization based gradient iterative algorithm, is proposed for rational models in this paper. The algorithm can obtain the optimal parameter estimates in one iteration for the reason that the information vectors obtained by using the Gram-Schmidt orthogonalization method are independent of each other. Compared to the least squares algorithm and the traditional gradient iterative algorithm, the proposed algorithm does not require the matrix inversion and eigenvalue calculation, thus it can be applied to nonlinear systems with complex structures or large-scale systems. Since the information vector of the rational models contains the latest output that is correlated with the noise, a biased compensation Gram-Schmidt orthogonalization based gradient iterative algorithm is introduced, by which the unbiased parameter estimates can be obtained. Two simulated examples are applied to demonstrate the efficiency of the proposed algorithm.

Key words: Rational model, gradient iterative algorithm, Gram-Schmidt orthogonalization, biased compensation, unbiased parameter estimate

1. Introduction

The major objective of the control engineering is to design a controller to force the dynamics of a control system to converge to the expectations of the track [1, 2, 3]. Robust controller designing usually has the assumption that the parameters of the control system should be known in prior. Therefore, system identification plays an important role in control engineering, and becomes a hot and challenging topic in research and applications [4, 5, 6]. There exist a plenty of identification algorithms, including the least squares algorithms [7, 8], the gradient iterative (GI) algorithms and the expectation maximization algorithms [9, 10, 11].

The least squares algorithms usually compute the parameter estimates by minimizing a quadratic cost function, thus a matrix inversion is involved [12, 13, 14]. In addition, the derivative function of the cost function must be analytic; otherwise, the parameter estimates may be unsolvable. Therefore, the LS algorithm will be inefficient for systems with more complex structures [15, 16]. The GI algorithm does not require to solve a function, so it can be used for nonlinear system identification [17, 18]. However, it has slower convergence rates because of its zigzagging nature. To increase the convergence rates, two ways are commonly considered: to choose a better direction and to compute a suitable step-length, e.g., the forgetting factor GI algorithm [19], the conjugate GI algorithm [20]. Once the direction is determined, the corresponding step-length should be calculated. Therefore, a challenging question exists: How to obtain an available step-length? In general, we...
should find the largest eigenvalue of the matrix. When the order of the matrix is large, the eigenvalue calculation is difficulty/impossible. That is, the traditional GI algorithm is inefficient for large-scale system identification.

Rational model widely exists in life science, for example, in bioengineering field [21], in chemical industry [22]. The rational model identification is more difficulty when compared with the polynomial nonlinear model [23], because the information vector of the rational model is correlated with the noise, which makes the parameter estimates biased [24, 25, 26]. Therefore, a biased compensation term should be involved to eliminate the biased term. For example, Chen et al proposed a biased compensation recursive least squares algorithm for rational models with time-delay [27]. Zhu et al provided an implicit LS algorithm for rational models, while the biased compensation term is constituted of the noise estimates [28]. The basic idea of the biased compensation based methods is to update the noise estimates in each iteration. Notice that the GI algorithm has slow convergence rates, using the initial inaccurate parameter estimates to compute the noise estimates may lead to divergence of the GI algorithm.

The reason why the GI algorithm has slower convergence rates is that the information vectors in information matrix are dependent on each other. Inspired by the Gram-Schmidt orthogonalization method [29, 30], we can make information vectors orthogonal to each other to improve the convergence rate of the GI algorithm. The focus of this paper is to propose a biased compensation Gram-Schmidt orthogonalization based GI algorithm for rational models, whose contributions are listed as follows:

1. The Gram-Schmidt orthogonalization based GI algorithm has quicker convergence rates (can get the optimal estimates in one iteration).
2. The Gram-Schmidt orthogonalization based GI algorithm is robust to the initial parameter estimates.
3. The Gram-Schmidt orthogonalization based GI algorithm does not require matrix inversion and eigenvalue calculation, so it can be applied to large-scale systems and nonlinear systems.
4. A biased compensation term is involved in each iteration which can get unbiased parameter estimates.

In summary, this paper is organised as follows. Section 2 introduces the rational model. Section 3 proposes a biased compensation GI algorithm. Section 4 develops a Gram-Schmidt orthogonalization based GI algorithm. Section 5 provides two simulation examples. Finally, Section 6 summarises the study and discusses future directions.

### 2. Rational model

The rational model is written by

\[ y(t) = \frac{a(t) + e(t)}{b(t)}, \]  

(1)

where \( a(t) \) and \( b(t) \) are functions whose structures are known in prior, \( e(t) \) is Gaussian white noise, and \( u(t) \) and \( y(t) \) are the input and output data, respectively.

Define

\[ a(t) = \varphi^T(t)\theta_a, \]

\[ b(t) = \psi^T(t)\theta_b. \]

The information vectors \( \varphi(t) = [\varphi_1(t), \cdots, \varphi_n(t)]^T \) and \( \psi(t) = [\psi_1(t), \cdots, \psi_m(t)]^T \) are constituted of the past inputs \( \{u(t-1), u(t-2), \cdots\} \) and the past outputs \( \{y(t-1), y(t-2), \cdots\} \) [24, 25, 26]. \( \theta_a \) and \( \theta_b \) are the unknown parameters defined as

\[ \theta_a = [a_1, a_2, \cdots, a_n]^T, \]  

(2)

\[ \theta_b = [b_1, b_2, \cdots, b_m]^T. \]  

(3)

Transforming the rational model into a regression model yields

\[ \hat{y}(t) = \varphi^T(t)\theta_a - y(t)\psi^T(t)\theta_b + b_1\psi_1(t)y(t) + e(t) \]
3. Biased compensation gradient iterative algorithm

To estimate the parameters \( \theta \) through the collected input-output data, the LS algorithms are widely used. However, the computational efforts of the LS algorithms increase intensively when the order \( m + n - 1 \) becomes larger \([31, 32]\). To deal with this problem, a biased compensation GI algorithm is proposed in this paper.

3.1. Traditional GI algorithm

Define the cost function

\[
J(\theta) = \frac{1}{2} [\tilde{Y}(N) - \Phi(N) \theta]^T [\tilde{Y}(N) - \Phi(N) \theta].
\]

The GI algorithm includes two steps: the first step is to determine an optimal direction which can make the estimates move toward the true values; and the second step is to compute a step-length along the direction.

Assume that the parameter vector estimate in iteration \( k - 1 \) is \( \hat{\theta}^{k-1} \), then we want to find a better parameter vector estimate \( \hat{\theta}^k \) to ensure

\[
J(\hat{\theta}^k) \leq J(\hat{\theta}^{k-1}).
\]

Let

\[
\hat{\theta}^k = \hat{\theta}^{k-1} + \lambda^{k-1} d^{k-1},
\]

where \( d^{k-1} \) is the direction whose corresponding step-length is \( \lambda^{k-1} \).

Since

\[
J(\hat{\theta}^k) = J(\hat{\theta}^{k-1}) + \frac{\partial J(\theta)}{\partial \theta} |_{\hat{\theta}^{k-1}} \lambda^{k-1} d^{k-1}.
\]

To keep \( J(\hat{\theta}^k) \leq J(\hat{\theta}^{k-1}) \), the best direction \( d^{k-1} \) is determined as

\[
d^{k-1} = -\frac{\partial J(\theta)}{\partial \theta} |_{\hat{\theta}^{k-1}} = \Phi^T(N)[\tilde{Y}(N) - \Phi(N) \hat{\theta}^{k-1}].
\]

According to \([30]\), the corresponding step-length \( \lambda^{k-1} \) can be computed by

\[
\lambda^{k-1} = \frac{1}{\lambda_{max} |\Phi^T(N)\Phi(N)|}.
\]

Then the traditional GI (T-GI) algorithm for estimating \( \theta \) is summarized as follows

\[
\hat{\theta}^k = \hat{\theta}^{k-1} + \frac{\Phi^T(N)}{\lambda_{max} |\Phi^T(N)\Phi(N)|} [\tilde{Y}(N) - \Phi(N) \hat{\theta}^{k-1}].
\]
3.2. Convergence property of the T-GI algorithm

Subtracting θ on both sides of Equation (7) yields

\[ \hat{θ}^k - θ = \hat{θ}^{k-1} - θ + \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} [Y(N) - Φ(N)\hat{θ}^{k-1}]. \] (8)

Let

\[ ε^k = \hat{θ}^k - θ, \]

Equation (8) can be transformed into

\[
ε^k = ε^{k-1} + \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} [Φ(N)θ + E(N) - Φ(N)\hat{θ}^{k-1}]
\]

\[ = ε^{k-1} - \frac{Φ^T(N)Φ(N)}{λ_{max}[Φ^T(N)Φ(N)]} ε^{k-1} + \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} E(N) \]

\[ = \left[ I - \frac{Φ^T(N)Φ(N)}{λ_{max}[Φ^T(N)Φ(N)]} \right] ε^{k-1} + \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} E(N). \]

Since \( e(t) \) is Gaussian white noise with zero mean, if

\[ E \left\{ \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} E(N) \right\} = 0, \]

then

\[ ε^k = \left[ I - \frac{Φ^T(N)Φ(N)}{λ_{max}[Φ^T(N)Φ(N)]} \right] ε^{k-1}, \]

which means

\[ \frac{∥ε^k∥}{∥ε^{k-1}∥} = \left\| \left[ I - \frac{Φ^T(N)Φ(N)}{λ_{max}[Φ^T(N)Φ(N)]} \right] \right\| < 1. \]

In this case, the GI algorithm is convergent. However, \( E(N) \) is correlated with the information matrix \( Φ(N) \). Thus the T-GI algorithm is biased.

Remark 1. Since the information matrix contains the current output \( y(t) \) which is correlated with the noise \( e(t) \), the T-GI algorithm is biased.

3.3. Biased compensation gradient iterative algorithm

To get the unbiased parameter estimates, the biased compensation GI algorithm is introduced.

In iteration \( k - 1 \), the parameter estimates are \( \hat{θ}^{k-1} \), the noise estimates then can be computed by

\[ ε^{k-1}(t) = \hat{y}(t) - \phi^T(t)\hat{θ}^{k-1}, \ t = 1, \cdots, N. \] (9)

The noise estimate vector can be expressed by

\[ \hat{E}^{k-1}(N) = [ε^{k-1}(1), \cdots, ε^{k-1}(N)]^T. \] (10)

Therefore, the noise estimates are applied to eliminate the biased term in the parameter estimates of the T-GI algorithm, and the biased compensation gradient iterative (BC-GI) algorithm is written by

\[ \hat{θ}^k = \hat{θ}^{k-1} + \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} [Y(N) - Φ(N)\hat{θ}^{k-1}] - \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} \hat{E}^{k-1}(N). \] (11)

In the BC-GI algorithm, the parameter estimation error \( ε^k \) is

\[ ε^k = ε^{k-1} + \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} [Y(N) - Φ(N)\hat{θ}^{k-1}] - \frac{Φ^T(N)}{λ_{max}[Φ^T(N)Φ(N)]} \hat{E}^{k-1}(N). \]
where

\[ N \]

the collected input-output data be \( \Phi(N) \).

### 4.1. Gram-Schmidt orthogonalization based gradient iterative algorithm

The BC-GI algorithm needs to compute the largest eigenvalue of a matrix. When the matrix order is large, the difficulty of calculation will increase accordingly. In addition, the BC-GI algorithm has slower convergence rates because of its zigzagging nature. In this section, we develop a Gram-Schmidt orthogonalization based gradient iterative (GO-GI) algorithm, which can avoid computing the eigenvalues and has quicker convergence rates.

**Remark 2.** In the BC-GI algorithm, the compensation term is introduced in each iteration to eliminate the biased term in the parameter estimates. Thus, the parameter estimates by using the BC-GI algorithm are unbiased.

In summary, the BC-GI algorithm constitutes of the following steps:

1. Let \( \hat{\theta}^0 = 1/p_0 \) with \( 1 \) being a column vector whose entries are all unity and \( p_0 = 10^6 \).
2. Collect the input-output data \( \{u(t), y(t)\} \), \( t = 1, \cdots, N \).
3. Form \( Y(N) \) and \( \Phi(N) \), and let \( k = 1 \).
4. Compute \( \hat{e}^{k-1}(t), t = 1, \cdots, N \).
5. Form \( \hat{E}^{k-1}(N) \).
6. Update \( \hat{\theta}^k \).
7. Compare \( \hat{\theta}^k \) with \( \hat{\theta}^{k-1} \); if \( \|\hat{\theta}^k - \hat{\theta}^{k-1}\| \leq \varepsilon \), then terminate the procedure and obtain \( \hat{\theta}^k \); otherwise, increase \( k \) by 1 and go to step 5).

### 4. Gram-Schmidt orthogonalization based gradient iterative algorithm

The BC-GI algorithm needs to compute the largest eigenvalue of a matrix. When the matrix order is large, the difficulty of calculation will increase accordingly. In addition, the BC-GI algorithm has slower convergence rates because of its zigzagging nature. In this section, we develop a Gram-Schmidt orthogonalization based gradient iterative (GO-GI) algorithm, which can avoid computing the eigenvalues and has quicker convergence rates.

#### 4.1. Gram-Schmidt orthogonalization based gradient iterative algorithm

In the GO-GI algorithm, each element in the parameter vector has its own direction and step-length. Let the collected input-output data be \( \Phi(N) \) and \( \hat{Y}(N) \), as mentioned in Section 2.

It follows that

\[
\hat{Y}(N) = \Phi(N)\hat{\theta} + E(N) = \omega(1)a_1 + \cdots + \omega(n)a_n + \omega(n + 1)b_2 + \cdots + \omega(n + m - 1)b_m + E(N),
\]

where

\[
\omega(1) = [\varphi_1(N), \varphi_1(N - 1), \cdots, \varphi_1(1)]^T \in \mathbb{R}^N,
\]

\[
\vdots
\]

\[
\omega(n) = [\varphi_n(N), \varphi_n(N - 1), \cdots, \varphi_n(1)]^T \in \mathbb{R}^N,
\]

\[
\omega(n + 1) = [-\psi_2(N)g(N), -\psi_2(N - 1)g(N - 1), \cdots, -\psi_2(1)g(1)]^T \in \mathbb{R}^N,
\]

\[
\vdots
\]
\[ \omega(n + m - 1) = [-\psi_m(N)y(N), -\psi_m(N-1)y(N-1), \ldots, -\psi_m(1)y(1)]^T \in \mathbb{R}^N. \]

Define the cost function
\[
J(\theta) = \frac{1}{2}[\bar{Y}(N) - \omega(1)a_1 - \cdots - \omega(n)a_n - \omega(n + 1)b_2 - \cdots - \omega(n + m - 1)b_m]^T
\times [\bar{Y}(N) - \omega(1)a_1 - \cdots - \omega(n)a_n - \omega(n + 1)b_2 - \cdots - \omega(n + m - 1)b_m]. \quad (13)
\]

Since \( \omega(i) \) is dependent on \( \omega(j) \) when \( i \neq j \), which means the parameter estimate \( a_i \) has an effect on the estimate \( a_j \). It leads to slow convergence rates. To overcome this difficulty, the Gram-Schmidt orthogonalization method is applied.

The new information vectors \( \mathbf{L}(j) \) are formed as
\[
\mathbf{L}(1) = \omega(1), \\
\vdots \\
\mathbf{L}(j) = \omega(j) - \sum_{i=1}^{j-1} \mathbf{L}(i)w_{i,j}, \quad i < j, \quad j = 2, 3, \ldots, n + m - 1,
\]
\[
w_{i,j} = \frac{\omega^T(j)\mathbf{L}(i)}{\mathbf{L}^T(i)\mathbf{L}(i)}.
\]

It follows that
\[
[\omega(1), \ldots, \omega(n + m - 1)] = [\mathbf{L}(1), \ldots, \mathbf{L}(n + m - 1)]\mathbf{W}, \quad (14)
\]
where
\[
\mathbf{W} = \begin{bmatrix}
1 & w_{1,2} & \cdots & w_{1,n+m-1} \\
0 & 1 & \cdots & w_{2,n+m-1} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1
\end{bmatrix}. \quad (15)
\]

The rational model is equivalent to
\[
\bar{Y}(N) = [\omega(1), \cdots, \omega(n), \omega(n + 1), \cdots, \omega(n + m - 1)]\theta + E(N)
\]
\[
= [\mathbf{L}(1), \cdots, \mathbf{L}(n + m - 1)]\mathbf{W}\theta + E(N).
\]

Let
\[
\mathbf{W}\theta = \alpha = [\alpha_1, \cdots, \alpha_{m+n-1}]^T.
\]

Then we have
\[
\bar{Y}(N) = [\mathbf{L}(1), \cdots, \mathbf{L}(m + n - 1)]\alpha + E(N). \quad (16)
\]

Define the cost function
\[
J(\alpha) = \frac{1}{2}[\bar{Y}(N) - (\mathbf{L}(1), \cdots, \mathbf{L}(m + n - 1))\alpha]^T[\bar{Y}(N) - (\mathbf{L}(1), \cdots, \mathbf{L}(m + n - 1))\alpha]
\]
\[
= \frac{1}{2}[\bar{Y}(N) - \mathbf{L}(1)\alpha_1 - \cdots - \mathbf{L}(m + n - 1)\alpha_{m+n-1}]^T[\bar{Y}(N) - \mathbf{L}(1)\alpha_1 - \cdots - \mathbf{L}(m + n - 1)\alpha_{m+n-1}].
\]

Assume that the parameter estimates in iteration \( k - 1 \) are \( \hat{\alpha}^{k-1} \). The purpose of the GO-GI algorithm is to find a new parameter vector estimate \( \hat{\alpha}^k \)
\[
\hat{\alpha}^k = \hat{\alpha}^{k-1} + \gamma^{k-1}d^{k-1}, \quad (17)
\]

which can ensure
\[
J(\hat{\alpha}^k) \leq J(\hat{\alpha}^{k-1}).
\]
Based on Equation (17), we have

\[ J(\hat{\alpha}^k) = J(\hat{\alpha}^{k-1} + \gamma^{k-1} d^{k-1}) \approx J(\hat{\alpha}^{k-1}) + \frac{\partial J(\alpha)}{\partial \alpha} \bigg|_{\hat{\alpha}^{k-1}} \gamma^{k-1} d^{k-1}. \]

The step-lengths \( \gamma^{k-1} \) are greater than zero, to keep \( J(\hat{\alpha}^k) \leq J(\hat{\alpha}^{k-1}) \), the following inequality should hold

\[ \frac{\partial J(\alpha)}{\partial \alpha} \bigg|_{\hat{\alpha}^{k-1}} d^{k-1} \leq 0. \]

The optimal direction is computed by

\[ d^{k-1} = \frac{\partial J(\alpha)}{\partial \alpha} \bigg|_{\hat{\alpha}^{k-1}}. \]

For the reason that \( L(i) \) is orthogonal to \( L(j), i \neq j \), we take the first-order functional derivative of \( J(\alpha) \) with respect to \( \alpha_1, \cdots, \alpha_{m+n+1} \) respectively to obtain

\[ d^{k-1} = [L^T(1)Y(N) - L^T(1)L(1)\alpha_1^{k-1}, \cdots, L^T(m + n - 1)Y(N) - L^T(m + n - 1)L(1)\alpha_{m+n+1}^{k-1}]^T. \]

Therefore, the parameter estimates \( \hat{\alpha}^k \) can be updated by

\[ \hat{\alpha}^k = \hat{\alpha}^{k-1} + \begin{bmatrix} \gamma_1^{k-1} & 0 & \cdots & 0 \\ 0 & \gamma_2^{k-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_{m+n+1}^{k-1} \end{bmatrix} d^{k-1}, \]

and the cost function \( J(\hat{\alpha}^k) \) is rewritten by

\[ J(\hat{\alpha}^k) = \frac{1}{2} [Y(N) - (L(1), \cdots, L(m + n - 1))\hat{\alpha}^k]^T [Y(N) - (L(1), \cdots, L(m + n - 1))\hat{\alpha}^k]. \]

Substitute Equation (20) into the above equation and take the first-order functional derivative of \( J(\gamma_1, \cdots, \gamma_{m+n+1}) \) with respect to \( \gamma_1, \cdots, \gamma_{m+n+1} \), respectively. It gives rise to

\[ \gamma_i^{k-1} = \frac{1}{|L^T(i)L(i)|}, \quad i = 1, 2, \cdots, m + n - 1. \]

Equation (20) can be simplified as

\[ \hat{\alpha}^k = \hat{\alpha}^{k-1} + \begin{bmatrix} L^T(1)Y(N) - L^T(1)L(1)\alpha_1^{k-1} \\ L^T(2)Y(N) - L^T(2)L(2)\alpha_2^{k-1} \\ \vdots \\ L^T(m + n - 1)Y(N) - L^T(m + n - 1)L(1)\alpha_{m+n+1}^{k-1} \end{bmatrix}. \]

**Remark 3.** Unlike the least squares algorithm in [33], the GO-GI algorithm avoids the matrix inversion (only calculate \( L^T(i)L(i) \)). Thus it has less computational efforts.

**Remark 4.** Once the parameter estimates \( \hat{\alpha}^k \) are obtained, the original parameter estimates \( \hat{\gamma}^k \) can be computed by

\[ \hat{\gamma}^k = [W]^{-1}\hat{\alpha}^k. \]
4.2. Properties of the GO-GI algorithm

In order to show the effectiveness of the GO-GI algorithm, some properties of the GO-GI algorithm are given in this subsection. Based on Equation (21), we have

\[
\begin{bmatrix}
\alpha_k^1 \\
\alpha_k^2 \\
\vdots \\
\alpha_{m+n-1}^k
\end{bmatrix} = \begin{bmatrix}
\alpha_1^{k-1} \\
\alpha_2^{k-1} \\
\vdots \\
\alpha_{m+n-1}^{k-1}
\end{bmatrix} + \begin{bmatrix}
[L^T(1)L(1)]_1 & 0 & \cdots & 0 \\
0 & [L^T(2)L(2)] & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & [L^T(m+n-1)L(m+n-1)]
\end{bmatrix} \times \\
\begin{bmatrix}
L^T(1)\bar{Y}(N) - L^T(1)L(1)\alpha_k^{k-1} \\
L^T(2)\bar{Y}(N) - L^T(2)L(2)\alpha_k^{k-1} \\
\vdots \\
L^T(m+n-1)\bar{Y}(N) - L^T(m+n-1)L(m+n-1)\alpha_k^{k-1}
\end{bmatrix}
\]

(22)

Equation (22) shows that the elements in the parameter vector can be estimated separately by using the GO-GI algorithm.

**Remark 5.** Compared with the T-GI algorithm [34], the GO-GI algorithm does not require the eigenvalue calculation, thus can be applied to large-scale system identification.

**Remark 6.** Equation (22) shows that the parameter estimates in iteration \( k \) are independent of the estimates in \( k-1 \), which indicates that the estimates obtained by using the GO-GI algorithm can obtain the local optimal values in one iteration, regardless of the initial parameter estimates.

**Remark 7.** Assume that the noise \( E(N) \) is uncorrelated with the information matrix \( \Phi(N) \), using the traditional least squares (LS) algorithm for the considered model, the parameter estimates are

\[
\hat{\vartheta} = [\Phi^T(N)\Phi(N)]^{-1}\Phi^T(N)\bar{Y}(N),
\]

it means that the LS algorithm also can get the local optimal values in one iteration. However, the computational efforts of the LS algorithm are heavier than those of the GO-GI algorithm, see Table 1 (only count the amount of multiplication and division).

Substituting Equation (16) into Equation (22) yields

\[
\begin{bmatrix}
\alpha_k^1 \\
\alpha_k^2 \\
\vdots \\
\alpha_{m+n-1}^k
\end{bmatrix} = \begin{bmatrix}
L^T(1)[L(1)\alpha_1 + L(2)\alpha_2 + \cdots + L(m+n-1)\alpha_{m+n-1} + E(N)] \\
L^T(2)[L(1)\alpha_1 + L(2)\alpha_2 + \cdots + L(m+n-1)\alpha_{m+n-1} + E(N)] \\
\vdots \\
L^T(m+n-1)[L(1)\alpha_1 + L(2)\alpha_2 + \cdots + L(m+n-1)\alpha_{m+n-1} + E(N)]
\end{bmatrix}.
\]

(23)

Since \( \mathbf{L}(i) \) is orthogonal to \( \mathbf{L}(j) \), \( i \neq j \), the above equation can be simplified to

\[
\begin{bmatrix}
\alpha_k^1 \\
\alpha_k^2 \\
\vdots \\
\alpha_{m+n-1}^k
\end{bmatrix} = \begin{bmatrix}
\alpha_1 + L^T(1)E(N) \\
\alpha_2 + L^T(2)E(N) \\
\vdots \\
\alpha_{m+n-1} + L^T(m+n-1)E(N)
\end{bmatrix}.
\]

(24)

The vectors \( \mathbf{L}(i), i = 1, \cdots, n \) are uncorrelated with the noise \( E(N) \), then

\[
E \begin{bmatrix}
L^T(i)E(N) \\
L^T(i)L(i)
\end{bmatrix} = 0, i = 1, \cdots, n.
\]
Table 1: The computational efficiency of LS and GO-GI algorithms \((M = m + n - 1)\)

| Algorithms | Step | Expressions | Flops |
|------------|------|-------------|-------|
| LS         | 1    | \(\omega(j) = \sum_{i=1}^{j-1} L(i) w_{i,j}\) | \(N(M^2 - 1) + \frac{1}{2}M(M - 1)\) |
|            | 2    | \(L^T(j)Y(N)\) | \((N+1)M\) |
| GO-GI      | 3    | \(W^{-1}\) | \(M^2\) |
|            | 4    | \(\hat{\alpha}^k = [W]^{-1}\hat{\alpha}^k\) | \(M^2\) |
| Sum        |      | \(F_1 := M^2N + \frac{5}{2}M^2 + \frac{1}{2}M + MN - N\) | |}

However, the vectors \(L(i), i = n + 1, \ldots, m + n - 1\) are correlated with the noise \(E(N)\), because the elements in \(L(i), i = n + 1, \ldots, m + n - 1\) contain \(y(t)\) which is correlated with \(e(t)\). Therefore, the last \(m-1\) estimates in the \([\alpha_1^k, \alpha_2^k, \ldots, \alpha_n^k, \alpha_{n+1}^k, \ldots, \alpha_{m+n-1}^k]\) are biased. To overcome this difficulty, we also use the biased compensation based method, that is:

\[
\begin{bmatrix}
\alpha_1^k \\
\vdots \\
\alpha_n^k \\
\alpha_{n+1}^k \\
\vdots \\
\alpha_{m+n-1}^k
\end{bmatrix} = \begin{bmatrix}
L^T(1)Y(N) \\
\vdots \\
L^T(n)Y(N) \\
L^T(n+1)Y(N) - L^T(n+1)\hat{\alpha}^{k-1} \\
\vdots \\
L^T(n+m-1)Y(N) - L^T(n+m-1)\hat{\alpha}^{k-1}
\end{bmatrix},
\]

where the noises are estimated by the parameter estimates \(\hat{\alpha}^{k-1}\),

\[
\hat{E}^{k-1}(N) = Y(N) - [L(1), \ldots, L(m + n - 1)]\hat{\alpha}^{k-1}.
\]

**Remark 8.** Equation (25) shows that the GO-GI algorithm can get the optimal parameter estimates in one iteration if the noise is Gaussian white and is uncorrelated with the information vector.

**Remark 9.** Since the parameter estimates are not accurate at first, the noise estimates based on the parameter estimates are also inaccurate. However, as iteration progresses, the errors between the estimates and the true values will be diminished. Thus, these two kinds of estimates will eventually approach to their true values.

5. Examples

5.1. Example 1

Consider the rational model proposed in [27],

\[
y(t) = \frac{0.2y(t-1) + 0.1y(t-1)u(t-1) + u(t-1) + e(t)}{1 + y^2(t-1) + y^2(t-2)}.\]


Then one can get
\[ y(t) = 0.2y(t-1) + 0.1y(t-1)u(t-1) + u(t-1) - y(t)y^2(t-1) - y(t)y^2(t-2) + e(t), \]
the \{u(t)\} satisfies \(N(0,1)\), \{e(t)\} is a white noise satisfies \(N(0,0.1^2)\), the number of the input-output data is \(N = 1000\). The simulation data are depicted in Figure 1.

![Figure 1: The simulation data](image)

Applying the T-GI and BC-GI algorithms to the rational model, the parameter estimates and their errors \(\tau = \|\hat{\vartheta}^k - \vartheta\|/||\vartheta||\) are shown in Figure 2 and Table 2.

| Algorithms | \(t\) | \(a_1\) | \(a_2\) | \(a_3\) | \(b_2\) | \(b_3\) | \(\tau\) (%) |
|------------|------|--------|--------|--------|-------|-------|----------|
| T-GI       | 10   | -0.01257 | -0.07729 | 0.54005 | 0.02794 | 0.01370 | 85.04295 |
|            | 50   | 0.02057  | -0.23455 | 0.82532 | 0.03485 | 0.48091 | 67.15774 |
|            | 100  | 0.05370  | -0.27025 | 0.92387 | 0.07698 | 0.69019 | 60.38739 |
|            | 150  | 0.07039  | -0.26113 | 0.95304 | 0.13428 | 0.75962 | 56.00576 |
|            | 200  | 0.08100  | -0.24075 | 0.96293 | 0.19208 | 0.78964 | 52.12342 |
|            | 250  | 0.08932  | -0.21861 | 0.96734 | 0.24668 | 0.80792 | 48.56001 |
|            | 300  | 0.09657  | -0.19729 | 0.97010 | 0.29735 | 0.82216 | 45.27933 |
| BC-GI      | 10   | 0.10122  | 0.04761 | 0.58562 | 0.01968 | 0.04566 | 82.10352 |
|            | 50   | 0.14371  | 0.07199 | 0.76416 | 0.42448 | 0.47474 | 46.75371 |
|            | 100  | 0.17068  | 0.08725 | 0.87888 | 0.69942 | 0.73724 | 23.95923 |
|            | 150  | 0.18357  | 0.09450 | 0.93336 | 0.83151 | 0.83055 | 13.12905 |
|            | 200  | 0.18971  | 0.09795 | 0.95926 | 0.89441 | 0.91906 | 7.98950 |
|            | 250  | 0.19262  | 0.09959 | 0.97157 | 0.92432 | 0.94687 | 5.55517 |
|            | 300  | 0.19401  | 0.10037 | 0.97743 | 0.93854 | 0.96009 | 4.40395 |
| True Values| 0.20000 | 0.10000 | 1.00000 | 1.00000 | 1.00000 |}

Use the GO-GI algorithm for the rational models. The parameter estimates and their errors are shown in
Table 3: The parameter estimates and errors (GO-GI)

|   | $a_1$ | $a_2$ | $a_3$ | $b_2$ | $b_3$ | $\tau$ (%) |
|---|-------|-------|-------|-------|-------|------------|
| 1 | 0.19500 | 0.11498 | 0.97544 | 0.95113 | 0.93368 | 5.00456    |
| 2 | 0.19500 | 0.11498 | 0.97544 | 0.95113 | 0.93368 | 5.00456    |
| 4 | 0.19500 | 0.11498 | 0.97544 | 0.95113 | 0.93368 | 5.00456    |
| True Values | 0.20000 | 0.10000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 |

Finally, apply the BC-GO-GI algorithm for the rational models. The parameter estimates and their errors are shown in Table 4.

Table 4: The parameter estimates and errors (BC-GO-GI)

|   | $a_1$ | $a_2$ | $a_3$ | $b_2$ | $b_3$ | $\tau$ (%) |
|---|-------|-------|-------|-------|-------|------------|
| 1 | 0.20017 | 0.11951 | 1.00004 | 1.00000 | 1.00000 | 1.11724    |
| 2 | 0.20017 | 0.11951 | 1.00004 | 1.00000 | 1.00000 | 1.11724    |
| 4 | 0.20017 | 0.11951 | 1.00004 | 1.00000 | 1.00000 | 1.11724    |
| True Values | 0.20000 | 0.10000 | 1.00000 | 1.00000 | 1.00000 | 1.00000 |

The element relative errors in the parameter estimates by using T-GI, BC-GI, GO-GI and BC-GO-GI algorithms with different iterations ($k = 2, 3, 5$) are shown in Figure 3. Assume that the output estimates between $[1001, 1200]$ are $\hat{y}(t)$. Using the parameter estimates $\hat{\theta}^{1000}$ to predict the outputs yields

$$\hat{y}(t) = \phi^T(t)\hat{\theta}^{1000}, \ t = 1000, 1001, \cdots, 1200,$$

while the true outputs are generated by

$$y(t) = 0.2y(t-1) + 0.1y(t-1)u(t-1) + u(t-1) - y(t)g^2(t-1) - y(t)g^2(t-2) + e(t), \ t = 1000, 1001, \cdots, 1200.$$
The output estimation errors $\delta = y(t) - \hat{y}(t)$ are shown in Figure 4.

Then the following findings can be obtained:

1. Figure 2 and Table 2 show that the BC-GI algorithm is more effective than the T-GI algorithm;
2. Tables 3 and 4 demonstrate that the BC-GO-GI algorithm has more accurate parameter estimates;
3. Tables 3 and 4 illustrate that the GO-GI algorithm can obtain the optimal estimates in one iteration;
4. Tables 4 and 2 show that the BC-GO-GI algorithm is more efficient than the BC-GI algorithm;
Figures 3 and 4 demonstrate that the BC-GO-GI is the most effective algorithm among these four algorithms.

5.2. Example 2

Consider the following Michaelis-Menten model in [35],

\[ y(t) = \frac{\beta}{1 + \frac{\alpha}{u(t)}} + v(t) \]

This model is usually applied to describe the enzyme kinetics. In such a model, \( y(t) \) is the initial velocity of an enzymatic reaction, \( u(t) \) is the substrate concentration. The parameters are: \( \alpha = 0.0641 \) and \( \beta = 212.6837 \).

In simulation, the input \( \{u(t)\} \) satisfies \( u(t) \sim N(0,1) \), the noise satisfies \( v(t) \sim N(0,0.1^2) \). Apply the GO-GI and the BC-GO-GI algorithms to the Michaelis-Menten model. The parameter estimates and their estimation errors are shown in Tables 5 and 6.

Table 5: The parameter estimates and errors (GO-GI)

| k | \( \alpha \) | \( \beta \) | \( \tau \) (%) |
|---|---|---|---|
| 1 | 0.00105 | 205.88449 | 3.19700 |
| 2 | 0.00105 | 205.88449 | 3.19700 |
| 4 | 0.00105 | 205.88449 | 3.19700 |

True Values 0.06410 212.68370

Table 6: The parameter estimates and errors (BC-GO-GI)

| k | \( \alpha \) | \( \beta \) | \( \tau \) (%) |
|---|---|---|---|
| 1 | 0.00413 | 210.10705 | 1.21186 |
| 2 | 0.00413 | 210.10705 | 1.21186 |
| 4 | 0.00413 | 210.10705 | 1.21186 |

True Values 0.06410 212.68370

Tables 5 and 6 also show that the BC-GO-GI algorithm is more effective than the GO-GI algorithm.

6. Conclusions

This paper proposes a Gram-Schmidt orthogonalization based gradient iterative (GO-GI) algorithm for rational models. Each information vector in the information matrix is independent of others. Therefore, the GO-GI algorithm can obtain the optimal estimates in one iteration no matter what the initial parameter estimates are. Since some information vectors contain the current outputs which are correlated with the noises, the estimates of the GO-GI algorithm are biased. To overcome this difficulty, a BC-GO-GI algorithm is developed. The simulation examples demonstrate that the BC-GO-GI algorithm is effective.

The GO-GI algorithm presented in this paper can avoid matrix inversion calculations and requires less computation with a quicker convergence rate. Therefore, it is believed that this study helps to make the GI algorithm an attractive choice for system identification, and may bring forward a new direction in using GO-GI algorithm for different kinds of systems.

**Conflict of interest** The authors declared that they have no conflicts of interest to this work.
References

[1] X. Tang, Y. Shi, L. Wang, A new framework for solving fractional optimal control problems using fractional pseudospectral methods, Automatica 78 (2017) 333-340.

[2] J. Na, Y.F. Lv, et al., Adaptive identifier-critic-based optimal tracking control for nonlinear systems with experimental validation, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020). DOI: 10.1109/TSMC.2020.3003224

[3] R. Doraiswami, L. Cheded, Robust Kalman filter-based least squares identification of a multivariable system, IET Control Theory and Applications 12 (8) (2018) 1064-1074.

[4] T. Soderstrom, U. Soverini, Errors-in-variables identification using maximum likelihood estimation in the frequency domain, Automatica 79 (2017) 131-143.

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

[6] C.P. Yu, L. Ljung, A. Wills, M. Verhaegen, Constrained subspace method for the identification of structured state-space models, IEEE Transactions on Automatic Control 65 (2020). doi:10.1109/TAC.2019.2957703

[7] G.Y. Chen, M. Gan, C.L.P. Chen, H.X. Li, A regularized variable projection algorithm for separable nonlinear least-squares problems, IEEE Transactions on Automatic Control 64 (2) (2019) 526-537.

[8] X. Zhang, Q.Y. Liu, et al., Recursive identification of bilinear time-delay systems through the redundant rule, Journal of the Franklin Institute 357 (1) (2020) 726-747.

[9] D.Q. Wang, S. Zhang, et al., A novel EM identification method for Hammerstein systems with missing output data, IEEE Transactions on Industrial Informatics 16 (4) (2020) 2500-2508.

[10] F. Ding, G. Liu, X.P. Liu, Partially coupled stochastic gradient identification methods for non-uniformly sampled systems, IEEE Transactions on Automatic Control 55 (8) (2010) 1976-1981.

[11] M.H. Li, X.M. Liu, et al. Filtering-based maximum likelihood iterative estimation algorithms for a special class of nonlinear systems with autoregressive moving average noise using the hierarchical identification principle, International Journal of Adaptive Control and Signal Processing 33 (7) (2019) 1189-1211.

[12] Q.Y. Liu, F. Ding, Auxiliary model-based recursive generalized least squares algorithm for multivariate output-error autoregressive systems using the data filtering, Circuits Systems and Signal Processing 38 (2) (2019) 590-610.

[13] J. Vörös, Modeling and identification of systems with backlash, Automatica 46 (2) (2010) 369-374.

[14] D.Q. Wang, L.W. Li, Y. Ji, Y.R. Yan, Model recovery for Hammerstein systems using the auxiliary model based orthogonal matching pursuit method, Applied Mathematical Modelling 54 (2018) 537-550.

[15] M. Gan, G.Y. Chen, et al., Term selection for a class of nonlinear separable models, IEEE Transactions on Neural Networks and Learning Systems (2020). DOI: 10.1109/TNNLS.2019.2904952

[16] H. Xu, F. Ding, E.F. Yang, Modeling a nonlinear process using the exponential autoregressive time series model, Nonlinear Dynamics 95 (3) (2019) 2079-2092.

[17] D.Q. Wang, L. Mao, et al., Recasted models based hierarchical extended stochastic gradient method for MIMO nonlinear systems, IET Control Theory and Applications 11 (4) (2017) 476-485.

[18] Y.W. Mao, F. Ding, et al., Highly efficient parameter estimation algorithms for Hammerstein nonlinear systems, IET Control Theory and Applications 13 (4) (2019) 477-485.

[19] J.X. Ma, W.L. Xiong, et al., Data filtering based forgetting factor stochastic gradient algorithm for Hammerstein systems with saturation and preload nonlinearities, Journal of the Franklin Institute 353 (16) (2016) 4280-4299.

[20] S. Abbasbandy, A. Jafarian, R. Ezzati, Conjugate gradient method for fuzzy symmetric positive definite system of linear equations, Applied Mathematics Computation 171 (2) (2005) 1184-1191.

[21] Klipp, E., Herwig, R., Kowald, A., Systems biology in practice: concepts, implementation and application, Weinheim, Germany: Wiley-VCH 2005

[22] Kamenski, D.I., Dimitrov, S.D., Parameter estimation in differential equations by application of rational functions, Computers & Chemical Engineering 17 (7) (1993) 643-651.

[23] J. Chen, Q.M. Zhu, M.F. Hu, L.X. Guo, P. Narayan, Improved gradient descent algorithms for time-delay rational state-space systems: intelligent search method and momentum method, Nonlinear Dynamics 101 (1) (2020) 361-373.

[24] S.A. Billings, Q.M. Zhu, Rational model identification using extended least squares algorithm, International Journal of Control 54 (3) (1991) 529-546.

[25] Q.M. Zhu, Y.J. Wang, D.Y. Zhao, S.Y. Li, S.A. Billings, Review of rational (total) nonlinear dynamic system modelling, identification and control, International Journal of Systems Science 46 (12) (2015) 2122-2133.

[26] B.Q. Mu, E.W. Bai, W.X. Zheng, Q.M. Zhu, A globally consistent nonlinear least squares estimator for identification of nonlinear rational systems, Automatica 77 (2017) 322-335.
[27] J. Chen, Q.M. Zhu, J. Li, Y.J. Liu, Biased compensation recursive least squares-based threshold algorithm for time-delay rational models via redundant rule, Nonlinear Dynamics 91 (2) (2018) 797-807

[28] Q.M. Zhu, An implicit least squares algorithm for nonlinear rational model parameter estimation, Applied Mathematical Modelling 29 (7) (2005) 673-689.

[29] D.Q. Wang, Y.R. Yan, Y.J. Liu, J.H. Ding, Model recovery for Hammerstein systems using the hierarchical orthogonal matching pursuit method, Journal of Computational and Applied Mathematics 345 (2019) 135-145.

[30] J. Chen, F. Ding, Y.J. Liu, Q.M. Zhu, Multi-step-length gradient iterative algorithm for equation-error type models, Systems & Control Letters 115 (2018) 15-21.

[31] L. Xu, F. Ding, Q.M. Zhu, Hierarchical Newton and least squares iterative estimation algorithm for dynamic systems by transfer functions based on the impulse responses, International Journal of Systems Science 50 (1) (2019) 141-151.

[32] F. Ding, L. Xu, F.E. Alsaadi, T. Hayat, Iterative parameter identification for pseudo-linear systems with ARMA noise using the filtering technique, IET Control Theory and Applications 12 (7) (2018) 892-899

[33] F. Ding, H.B. Chen, et al., A hierarchical least squares identification algorithm for Hammerstein nonlinear systems using the key term separation, Journal of the Franklin Institute 355 (8) (2018) 3737-3752.

[34] Z.W. Ge, F. Ding, et al., Gradient-based iterative identification method for multivariate equation-error autoregressive moving average systems using the decomposition technique, Journal of the Franklin Institute 356 (3) (2019) 1658-1676.

[35] J. Chen, Y.J. Liu, Q.M. Zhu, A bias compensation recursive algorithm for dual-rate rational models, IET Control Theory and Applications, 12 (16) (2018) 2184-2193.