Highly efficient maximum-likelihood identification methods for bilinear systems with colored noises

Meihang Li · Ximei Liu · Yamin Fan

Received: 8 April 2021

Abstract As a special class of nonlinear systems, bilinear systems can naturally describe many industrial production process. This paper mainly discussed the highly efficient iterative identification methods for bilinear systems with autoregressive moving average noise. Firstly, the input-output representation of the bilinear systems is derived through eliminating the unknown state variables in the model. Then based on the maximum-likelihood principle and the negative gradient search principle, a maximum-likelihood gradient-based iterative (ML-GI) algorithm is proposed to identify the parameters of the bilinear systems with colored noises. For further improving the computational efficiency, the original identification model is divided into three sub-identification models with smaller dimensions and fewer parameters, and a hierarchical maximum-likelihood gradient-based iterative (H-ML-GI) algorithm is derived by using the hierarchical identification principle. A gradient-based iterative (GI) algorithm is given for comparison. Finally, the algorithms are verified by a simulation example. The simulation results show that the proposed algorithms are effective for identifying bilinear systems with colored noises and the H-ML-GI algorithm has a higher computational efficiency and a faster convergence rate than the ML-GI algorithm and the GI algorithm.

Keywords Parameter estimation · Iterative identification · Gradient search · Maximum-likelihood · Bilinear system

1 Introduction

Nonlinear systems can be found in many practical problems and have many characteristics which are significantly different from linear systems [1–3]. The modelling and parameter identification of nonlinear systems are much difficult because of their complexity [4,5]. Bilinear systems are the simplest nonlinear systems in the form [6], and can naturally describe many industrial production process, especially the chemical process [7–9]. They are jointly linear in the state and the force variables, and are the simple nonlinear extension of linear systems [10]. Therefore, bilinear systems are considered to be nonlinear systems that are closest to linear systems. Because of the widely applications of bilinear systems, the study of their identification methods is meaningful.

The parameter identification of bilinear systems has been studied for decades and many identification methods have been proposed for bilinear systems [11,12]. At the early time, many classical identification methods were used to identify bilinear systems, such as Walsh functions,
block-pulse functions, Chebyshev polynomials, Taylor polynomials and Galerkin methods. Favoreel et al. generalized linear subspace identification theory to the subspace identification of bilinear systems [13]. However, the size of the Hankel matrix increases exponentially with the number of the dimensionality of the system states and system inputs-outputs. In addition, some parameter identification methods of bilinear systems are based on the conversion of the bilinear system into an equivalent linear model, such as the Kalman filtering-based identification algorithm [14], the superspace method [15] and the intersection subspace algorithm [16]. Besides, the least squares methods [17,18], the gradient methods [19,20], the maximum-likelihood methods [21–23] and the intelligent algorithms [24] have been developed to identify the bilinear systems.

As a class of important approaches for dynamical system identification, the maximum-likelihood estimation methods have good statistical properties and have been applied in image recognition, magnetic resonance, speech recognition, bioinformatics, and system identification, etc [25,26]. For example, Chen et al. presented a parameter separation least squares-based iterative algorithm and a maximum-likelihood least squares-based iterative method for the output-error bilinear-parameter models with colored noises [27]; Alfonsi et al. studied the maximum-likelihood estimator for estimating the drift parameters of a Wishart process [28]; Aslam derived a recursive maximum-likelihood least squares-based secondary path identification algorithm for active noise control systems with autoregressive moving average noise [29]; Xia et al. derived a key term separation-based maximum-likelihood recursive extended stochastic gradient algorithm and a key term separation-based maximum-likelihood multi-innovation extended stochastic gradient algorithm for the multi-variable input nonlinear systems with unmeasured disturbances [30]; Gibson et al. proposed a new expectation maximisation-based maximum-likelihood algorithm and a gradient-based algorithm for parameter estimation of multi-variable bilinear systems [31].

As an optimization algorithm, the gradient method is useful in the field of identification [32, 33]. Many gradient-based parameter identification methods have been developed using the auxiliary model approach, the multi-innovation identification theory, the hierarchical identification principle and the data filtering technique [34–36], etc. For example, Chen et al. developed an Aitken-based modified Kalman filtering stochastic gradient algorithm for dual-rate nonlinear models [37]; a multi-innovation stochastic gradient parameter estimation method and a recursive least squares algorithm were derived for parameter estimation of the sine combination signals and periodic signals in [38].

The maximum-likelihood estimation algorithm of bilinear systems in [21] is recursive and in [31] is based on the expectation maximisation method. Based on the previous work [39] and different from them, this paper focuses on the highly efficient parameter identification problems of bilinear systems with ARMA noise based on the maximum-likelihood principle [40] and the hierarchical identification principle. The main contributions of this paper are as follows.

- The model structure of bilinear state space systems involves the products of the states and inputs, which increases the difficulty of system identification. This paper eliminates the state variable in the systems and obtains the input-output representation of the bilinear state space system, and derives a maximum-likelihood gradient-based iterative algorithm by combing the maximum-likelihood principle with the negative gradient search principle.
- For further improving the computational efficiency, the original identification model is divided into three sub-identification models with smaller dimensions and fewer parameters, and a hierarchical maximum-likelihood gradient-based iterative algorithm is developed based on the hierarchical identification principle. Then a gradient-based iterative algorithm is presented for comparison.

The remains of this paper is organized as follows. The identification model of bilinear systems with colored noises is derived in Section 2. Section 3 discusses the maximum-likelihood principle and proposes a maximum-likelihood gradient-based iterative algorithm. For further improving the computational efficiency, a hierarchical maximum-likelihood gradient-based iterative algorithm is proposed in Section 4. For comparison, a gradient-based iterative identification algorithm is presented in Section 5. The computational efficiency of the algorithms are discussed in Section 6. A simulation example is provided in Section 7 to verify the effectiveness of the proposed algorithms. Finally, we make some concluding remarks in Section 8.
2 System description and identification model

Let us introduce some notation. "A := X" or "X := A" stands for "A is defined as X". \(1_n\) represents an \(n\)-dimensional column vector whose elements are 1. Let \(T\) be the matrix transpose. \(r^{-1}\) stands for a unit backward shift operator: \(r^{-1}y(s) = y(s - 1)\).

Consider the bilinear state space system in Figure 1, and assume that the bilinear system has the following observability canonical form [17]:

\[
\begin{align*}
\dot{x}(s + 1) &= Ax(s) + Bx(s)u(s) + gu(s), \\
y(s) &= hx(s),
\end{align*}
\]

where \(x(s) := [x_1(s), x_2(s), \cdots, x_n(s)]^T\) is the \(n\)-dimensional state vector, \(u(s) \in \mathbb{R}\) and \(y(s) \in \mathbb{R}\) are the input and output of the system, respectively, and \(A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^{1 \times n}, g \in \mathbb{R}^n\) and \(h \in \mathbb{R}^{1 \times n}\) are constant matrices and vectors:

\[
A := \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \\
\end{bmatrix},
B := \begin{bmatrix}
0 \\
b \\
\end{bmatrix} \in \mathbb{R}^{n \times n},
\]

\[
b := [-b_1, -b_{n-1}, -b_{n-2}, \cdots, -b_1],
\]

\[
g := [g_1, g_2, \cdots, g_{n-1}, g_n]^T,
\]

\[
h := [1, 0, \cdots, 0, 0].
\]

Referring to the method in [17, 20], from (1)–(2), we can obtain the input-output representation of form:

\[
[A(r^{-1}) + u(s - n)B(r^{-1})]y(s) = [C(r^{-1}) + u(s - n)D(r^{-1})]u(s),
\]

where

\[
A(r^{-1}) := 1 + a_1r^{-1} + a_2r^{-2} + \cdots + a_nr^{-n}, \quad a_i \in \mathbb{R},
\]

\[
B(r^{-1}) := b_1r^{-1} + b_2r^{-2} + \cdots + b_nr^{-n}, \quad b_i \in \mathbb{R},
\]

\[
C(r^{-1}) := c_1r^{-1} + c_2r^{-2} + \cdots + c_nr^{-n}, \quad c_i \in \mathbb{R},
\]

\[
D(r^{-1}) := d_1r^{-2} + d_3r^{-3} + \cdots + d_nr^{-n}, \quad d_i \in \mathbb{R}.
\]

The coefficients \(a_i\) and \(d_i\) and the parameters \(a_i, b_i\) and \(g_i\) have the following relations:

\[
\begin{align*}
[c_1, \cdots, c_2, c_1] &:= [g_1 + a_n-1g_1 + a_n-2g_2 + \cdots + a_1g_{n-1}, \cdots, g_2 + a_1g_1, g_1] \in \mathbb{R}^{1 \times n}, \\
d_{n-1, \cdots, d_3, d_2} &:= [b_{n-1}g_1 + b_{n-2}g_2 + \cdots + b_1g_{n-1}, \cdots, b_1g_1] \in \mathbb{R}^{1 \times (n-1)}.
\end{align*}
\]

The practical industrial processes are always interfered with various factors, such as disturbances. These factors are generally called stochastic noise. Introducing a noise term \(w(s) \in \mathbb{R}\) to (3), we have

\[
[A(r^{-1}) + u(s - n)B(r^{-1})]y(s) = [C(r^{-1}) + u(s - n)D(r^{-1})]u(s) + w(s),
\]
where the noise term \( w(s) \) may be a white noise process, an autoregressive (AR) process, a moving average (MA) process or an ARMA process. Without loss of generality, this paper considers \( w(s) \) as an ARMA process, and the algorithms studied in this paper can be applied to other special cases with an MA noise and an AR noise.

This paper considers \( w(s) \) as an ARMA process, that is, \( E(r^{-1})w(s) = F(r^{-1})v(s) \), \( v(s) \in \mathbb{R} \) is a Gaussian distributed white noise with zero mean and variance \( \sigma^2 \). \( E(r^{-1}) \) and \( F(r^{-1}) \) are polynomials in \( r^{-1} \) and

\[
E(r^{-1}) := 1 + e_1 r^{-1} + e_2 r^{-2} + \cdots + e_n r^{-n_e} , \quad e_i \in \mathbb{R} ,
\]

\[
F(r^{-1}) := 1 + f_1 r^{-1} + f_2 r^{-2} + \cdots + f_n r^{-n_f} , \quad f_i \in \mathbb{R} .
\]

Assume that the system orders \( n_e, n_f \) and \( n_f \) are known and \( u(s) = 0, y(s) = 0 \) and \( v(s) = 0 \) as \( s \leq 0 \). The objective of this paper is to develop the highly efficient iterative algorithm for estimating the parameters \( a_i, b_i, c_i, d_i, e_i \) and \( f_i \) from the measured input-output data \( \{ u(i), y(i) : i = (s - 1)l + 1, (s - 1)l + 2, \cdots, sl \} \) (\( l \) denotes the data length). Define the parameter vector \( \vartheta \) and information vector \( \varphi(s) \) as

\[
\vartheta := \begin{bmatrix} \theta \\ \rho \end{bmatrix} \in \mathbb{R}^{4n+n_e+n_f-1} , \quad \varphi(s) := \begin{bmatrix} \phi(s) \\ \psi(s) \end{bmatrix} \in \mathbb{R}^{4n+n_e+n_f-1} ,
\]

where

\[
\theta := [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n, c_1, c_2, \cdots, c_n, d_2, d_3, \cdots, d_n]^T \in \mathbb{R}^{n-1} ,
\]

\[
\rho := [e_1, e_2, \cdots, e_{n_e}, f_1, f_2, \cdots, f_{n_f}]^T \in \mathbb{R}^{n_e+n_f} ,
\]

\[
\phi(s) := [-y(s-1), -y(s-2), \cdots, -y(s-n), -u(s-n)y(s-1), -u(s-n)y(s-2), \cdots, -u(s-n)u(s-3), \cdots, u(s-n)u(s-n)]^T \in \mathbb{R}^{4n-1} ,
\]

\[
\psi(s) := [-w(s-1), -w(s-2), \cdots, -w(s-n), v(s-1), v(s-2), \cdots, v(s-n)]^T \in \mathbb{R}^{n_e+n_f} .
\]

According to the above definitions, \( E(r^{-1})w(s) = F(r^{-1})v(s) \) can be written as

\[
w(s) = [1 - E(r^{-1})]w(s) + F(r^{-1})v(s) = \psi^T(s)\rho + v(s) .
\]

Equation (9) is the noise model.

Thus, Equation (4) can be written as

\[
y(s) = -\sum_{i=1}^{n} a_i y(s-i) - u(s-n) \sum_{i=1}^{n} b_i y(s-i) + \sum_{i=1}^{n} c_i u(s-i) + u(s-n) \sum_{i=2}^{n} d_i u(s-i) \\
-\sum_{i=1}^{n_e} e_i w(s-i) + \sum_{i=1}^{n_f} f_i v(s-i) + v(s)
\]

\[
= \phi^T(s)\theta + w(s) 
= \phi^T(s)\theta + \psi^T(s)\rho + v(s) 
= \varphi^T(s)\vartheta + v(s) .
\]

Equation (13) is the identification model for the bilinear system in (4), and the parameter vector \( \vartheta \) contains the parameter vector \( \theta \) of the system model and the parameter vector \( \rho \) of the noise model.

3 The maximum-likelihood gradient-based iterative algorithm

3.1 The maximum-likelihood principle

For the identification model in (13), the maximum-likelihood estimation of \( \vartheta \) is obtained by maximizing the likelihood function, or the probability density function of the observation data \( u_i = \{ u((s-1)l+1), u((s-1)l+2), \cdots, u(sl) \} \) and \( y_i = \{ y((s-1)l+1), y((s-1)l+2), \cdots, y(sl) \} \) conditioned on the parameter vector \( \vartheta \), that is,

\[
\hat{\vartheta}_{ML} = \arg \max_{\theta} p(y_i|u_{i-1}, \theta) .
\]
where \( l \) denotes the data length and \( l \gg 4n + n_c + n_f - 1 \).

Assume that the observed values \( \{u(i), y(i), i < s\} \) and the parameter vector \( \theta \) is independent with \( v(s) \), and the probability density function \( p(y | u_{i-1}, \theta) \) is available. Referring to the derivation process of the objective function in [39], the maximum-likelihood function is given by

\[
p(y | u_{i-1}, \theta) = \frac{1}{(2\pi \sigma^2)^{n/2}} \exp \left( - \frac{1}{2\sigma^2} \sum_{j=(s-1)l+1}^{sl} v^2(j) \right) + K,
\]

where \( K \) is a constant.

Taking the natural logarithm on both sides of (14) and maximizing the logarithm likelihood function gives the following equivalent cost function

\[
J(\theta) = \frac{1}{2} \sum_{j=(s-1)l+1}^{sl} v^2(j),
\]

where \( v(s) \) is given by

\[
v(s) = \frac{E(r^{-1})}{F(r^{-1})} \{ [A(r^{-1}) + u(s-n)B(r^{-1})]y(s) - [C(r^{-1}) + u(s-n)D(r^{-1})]u(s) \}.
\]

Obviously, \( J(\theta) \) and \( v(s) \) depend on the parameter vector \( \theta \), which contains the parameters \( c_i, b_1, c_i, d_i, e_i \) and \( f_i \). In the following, we use the iterative methods to obtain the minimum of \( J(\theta) \).

### 3.2 The iterative identification algorithm

Let \( \hat{\theta}_k(s) \) be the estimate of \( \theta \) at iteration \( k \), i.e.,

\[
\hat{\theta}_k(s) := \begin{bmatrix} \hat{\theta}_k(s) \\ \hat{\rho}_k(s) \end{bmatrix},
\]

\[
\hat{\theta}_k(s) := [\hat{a}_{1,k}(s), \hat{a}_{2,k}(s), \ldots, \hat{a}_{n,k}(s), \hat{b}_{1,k}(s), \hat{b}_{2,k}(s), \ldots, \hat{b}_{n,k}(s), \hat{c}_{1,k}(s), \hat{c}_{2,k}(s), \ldots, \hat{c}_{n,k}(s),
\]

\[
\hat{d}_{2,k}(s), \hat{d}_{3,k}(s), \ldots, \hat{d}_{n,k}(s) \}^T \in \mathbb{R}^{4n-1},
\]

\[
\hat{\rho}_k(s) := [\hat{e}_{1,k}(s), \hat{e}_{2,k}(s), \ldots, \hat{e}_{n,k}(s), \hat{f}_{1,k}(s), \hat{f}_{2,k}(s), \ldots, \hat{f}_{n,k}(s)] \in \mathbb{R}^{n_a + n_f}.
\]

Using the gradient search and minimizing the cost function \( J(\theta) \), we can obtain the gradient-based iterative parameter estimate \( \hat{\theta}_k(s) \) of \( \theta \):

\[
\hat{\theta}_k(s) := \hat{\theta}_{k-1}(s) - \mu_k(s) \text{grad} \left[ J \left( \hat{\theta}_{k-1}(s) \right) \right]
\]

\[
= \hat{\theta}_{k-1}(s) - \mu_k(s) \sum_{j=(s-1)l+1}^{sl} \hat{v}_k(j) \left. \frac{\partial v(j)}{\partial \theta} \right|_{\hat{\theta}_{k-1}(s)},
\]

where \( \mu_k(s) \) is an iterative step-size or a convergence factor.

Let \( \hat{A}_k(r^{-1}), \hat{B}_k(r^{-1}), \hat{C}_k(r^{-1}), \hat{D}_k(r^{-1}), \hat{E}_k(r^{-1}) \) and \( \hat{F}_k(r^{-1}) \) be the estimates of the polynomials \( A(r^{-1}), B(r^{-1}), C(r^{-1}), D(r^{-1}), E(r^{-1}) \) and \( F(r^{-1}) \) at iteration \( k \):

\[
\hat{A}_k(r^{-1}) := 1 + \hat{a}_{1,k}r^{-1} + \hat{a}_{2,k}r^{-2} + \cdots + \hat{a}_{n,k}r^{-n},
\]

\[
\hat{B}_k(r^{-1}) := \hat{b}_{1,k}r^{-1} + \hat{b}_{2,k}r^{-2} + \cdots + \hat{b}_{n,k}r^{-n},
\]

\[
\hat{C}_k(r^{-1}) := \hat{c}_{1,k}r^{-1} + \hat{c}_{2,k}r^{-2} + \cdots + \hat{c}_{n,k}r^{-n},
\]

\[
\hat{D}_k(r^{-1}) := \hat{d}_{2,k}r^{-2} + \hat{d}_{3,k}r^{-3} + \cdots + \hat{d}_{n,k}r^{-n},
\]

\[
\hat{E}_k(r^{-1}) := 1 + \hat{e}_{1,k}r^{-1} + \hat{e}_{2,k}r^{-2} + \cdots + \hat{e}_{n,k}r^{-n},
\]

\[
\hat{F}_k(r^{-1}) := 1 + \hat{f}_{1,k}r^{-1} + \hat{f}_{2,k}r^{-2} + \cdots + \hat{f}_{n,k}r^{-n}.
\]

Define the information vector \( \hat{\varphi}_{f,k}(s) \) as

\[
\hat{\varphi}_{f,k}(s) := - \left. \frac{\partial v(s)}{\partial \theta} \right|_{\hat{\theta}_{k-1}(s)}
\]
\[
\begin{align*}
\varphi_{f,k}(s) & = [-\hat{y}_{f,k}(s-1), -\hat{y}_{f,k}(s-2), \ldots, -\hat{y}_{f,k}(s-n), -u(s-n)\hat{y}_{f,k}(s-1), \\
& \quad -u(s-n)\hat{y}_{f,k}(s-2), \ldots, -u(s-n)\hat{y}_{f,k}(s-n), \hat{u}_{f,k}(s-1), \hat{u}_{f,k}(s-2), \ldots, \\
& \quad \hat{u}_{f,k}(s-n), u(s-n)\hat{u}_{f,k}(s-2), u(s-n)\hat{u}_{f,k}(s-3), \ldots, u(s-n)\hat{u}_{f,k}(s-n)],
\end{align*}
\]
The hierarchical maximum-likelihood gradient-based iterative algorithm of large-size matrix, which increases the computation burden of the algorithm. For further

\[ -\hat{w}_{f,k}(s-1), -\hat{w}_{f,k}(s-2), \ldots, -\hat{w}_{f,k}(s-n_e), \hat{v}_{f,k}(s-1), \hat{v}_{f,k}(s-2), \ldots, \]
\[ \hat{v}_{f,k}(s-n_f) \] \[ T. \]

(32)

Define the estimate of \( \varphi(s) \) as

\[ \hat{\varphi}_k(s) := [\varphi^T(s), \hat{\varphi}_k(s)]^T \in \mathbb{R}^{4n+n_e+n_f-1}, \]

(33)

\[ \hat{\psi}_k(s) := [-\hat{w}_{k-1}(s-1), -\hat{w}_{k-1}(s-2), \ldots, -\hat{w}_{k-1}(s-n_e), \hat{v}_{k-1}(s-1), \hat{v}_{k-1}(s-2), \ldots, \hat{v}_{k-1}(s-n_f)]^T \in \mathbb{R}^{n_e+n_f}. \]

(34)

From (13), we have \( v(s-j) = y(s-j) - \varphi^T(s-j)\theta \). Replacing \( \varphi(s) \) and \( \theta \) with their estimates \( \hat{\varphi}_k(s) \) and \( \hat{\theta}_k(s) \), the estimate of \( v(s) \) can be computed through

\[ \hat{v}_k(s-j) = y(s-j) - \hat{\varphi}_k^T(s-j)\hat{\theta}_k(s). \]

(35)

From (11), we have \( w(s-j) = y(s-j) - \theta^T(s-j)\theta \). Replacing \( \theta \) with \( \hat{\theta}_k(s) \) gives the estimate of \( w(s) \):

\[ \hat{w}_k(s-j) = y(s-j) - \hat{\theta}_k^T(s-j)\hat{\theta}_k(s). \]

(36)

Using the above definitions and (35) and from (20), we have

\[ \hat{\theta}_k(s) = \hat{\theta}_{k-1}(s) + \mu_k(s) \sum_{j=(s-1)+1}^{s} \hat{v}_k(j)\hat{\varphi}_{f,k}(j) \]
\[ = \hat{\theta}_{k-1}(s) + \mu_k(s) \sum_{j=(s-1)+1}^{s} \hat{\varphi}_{f,k}(j)[y(j) - \hat{\varphi}_k^T(j)\hat{\theta}_{k-1}(s)]. \]

(37)

In order to guarantee the convergence of \( \hat{\theta}_k(s) \), one conservative choice of \( \mu_k(s) \) is to satisfy

\[ 0 < \mu_k(s) \leq 2\lambda_{\max}^{-1} \left[ \sum_{j=(s-1)+1}^{s} \hat{\varphi}_{f,k}(j)\hat{\varphi}_{f,k}^T(j) \right]. \]

(38)

Equations (7), (17)–(19) and (28)–(38) form the maximum-likelihood gradient-based iterative (ML-GI) identification algorithm for the bilinear systems.

The identification steps of the ML-GI algorithm to compute \( \hat{\theta}_k(s) \) are listed as follows.

1. Let \( s = 1 \) and set \( \hat{\theta}_0(s) = 1_{4n+n_e+n_f-1}/p_0 \), give the data length \( l \), a small \( \varepsilon > 0 \) and a maximum iteration \( k_{\max} \).
2. Collect the input-output data \( \{u(i), y(i); i = (s-1)l + 1, (s-1)l + 2, \ldots, s l\} \).
3. Let \( k = 1 \), and set the initial values: \( \hat{v}_0(s-j) \) and \( \hat{v}_0(s-j) \) are random numbers, \( j = 1, 2, \ldots, \) \( \max[n_e, n_f] \). Set \( \hat{y}_{f,0}(s-i), \hat{w}_{f,0}(s-i) \) and \( \hat{w}_{f,0}(s-i) \) be random numbers, \( i = 1, 2, \ldots, \) \( \max[n_e, n_f] \).
4. Form \( \phi(s) \) using (7), form \( \psi_k(s) \) using (34), form \( \varphi_k(s) \) using (33).
5. Compute \( \hat{y}_{f,k}(s) \) using (28), \( \hat{u}_{f,k}(s) \) using (29), \( \hat{w}_{f,k}(s) \) using (30) and \( \hat{v}_{f,k}(s) \) using (31), and form \( \varphi_{f,k}(s) \) using (32).
6. Select a large \( \mu_k(s) \) satisfying (38), and update the estimate \( \hat{\theta}_k(s) \) using (37).
7. Read \( \hat{\theta}_k(s) \) from \( \hat{\theta}_{k-1}(s) \) in (17), and compute \( \hat{v}_k(s-j) \) using (36), \( \hat{v}_k(s-j) \) using (35).
8. If \( k < k_{\max} \), increase \( k \) by 1 and go to Step 4; otherwise, proceed to next step.
9. Compare \( \hat{\theta}_k(s) \) with \( \hat{\theta}_{k-1}(s) \), if \( \|\hat{\theta}_k(s) - \hat{\theta}_{k-1}(s)\| > \varepsilon \), set \( \hat{\theta}_0(s+1) = \hat{\theta}_k(s) \), increase \( s \) by 1 and go to Step 2; otherwise, obtain the parameter estimate \( \hat{\theta}_k(s) \) and terminate the calculation process.

The flowchart of computing \( \hat{\theta}_k(s) \) in the ML-GI algorithm is shown in Figure 2.

4 The hierarchical maximum-likelihood gradient-based iterative algorithm

The computation of the step-sizes in the ML-GI algorithm involves computing the eigenvalue of large-size matrix, which increases the computation burden of the algorithm. For further
Start and Initialize: let $s = 1$, set $l$, $\varepsilon$, $\hat{\vartheta}_0(s)$ and $k_{\text{max}}$

Collect the input-output data $\{u(i), y(i)\}$, let $k = 1$ and set initial values

Form $\phi(s)$, $\hat{\psi}_k(s)$ and $\hat{\varphi}_k(s)$

Compute $\hat{y}_{f,k}(s)$, $\hat{u}_{f,k}(s)$, $\hat{w}_{f,k}(s)$ and $\hat{v}_{f,k}(s)$, form $\hat{\varphi}_{f,k}(s)$ and select a large $\mu_k(s)$

Update the parameter estimate $\hat{\vartheta}_k(s)$

Compute $\hat{w}_k(s-j)$ and $\hat{v}_k(s-j)$

$k = k_{\text{max}}$?

No

Yes

$\|\hat{\vartheta}_k(s) - \hat{\vartheta}_{k-1}(s)\| \leq \varepsilon$?

No

Yes

$\hat{\vartheta}_0(s+1) := \hat{\vartheta}_k(s)$

Obtain the parameter estimates $\hat{\vartheta}_k(s)$

End

Fig. 2 The flowchart of the ML-GI algorithm

improving the computational efficiency, the original identification model is divided into three sub-
identification models with smaller dimensions and fewer parameters, and a hierarchical maximum-
likelihood gradient-based iterative algorithm is derived in this section by using the hierarchical
identification principle.

Define the parameter vector $\vartheta$ and information vector $\varphi(s)$ as

$$
\vartheta := \begin{bmatrix}
\theta_y \\
\theta_u \\
\rho
\end{bmatrix} \in \mathbb{R}^{4n+n_e+n_f-1}, \quad
\varphi(s) := \begin{bmatrix}
\phi_y(s) \\
\phi_u(s)
\end{bmatrix} \in \mathbb{R}^{4n+n_e+n_f-1},
$$

where

$$
\theta_y := [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n]^T \in \mathbb{R}^{2n},
$$

$$
\theta_u := [c_1, c_2, \cdots, c_n, d_2, d_3, \cdots, d_n]^T \in \mathbb{R}^{2n-1},
$$

$$
\phi_y(s) := [-y(s-1), -y(s-2), \cdots, -y(s-n), -u(s-n)y(s-1), -u(s-n)y(s-2), \cdots, -u(s-n)y(s-n), u(s-n)]^T \in \mathbb{R}^{2n},
$$

$$
\phi_u(s) := [u(s-1), u(s-2), \cdots, u(s-n), u(s-n)u(s-2), u(s-n)u(s-3), \cdots],
$$
where

\[ u(s - n)u(s - n)^T \in \mathbb{R}^{2n-1} , \]  

and the parameter vector \( \rho \) and the information vector \( \psi(s) \) are defined in (6) and (8). Then the identification model in (13) can be rewritten as

\[
y(s) = \phi_y^T(s)\theta_y + \phi_u^T(s)\theta_u + w(s) = \phi_y^T(s)\theta_y + \phi_u^T(s)\theta_u + \psi^T(s)\rho + v(s). \tag{43}
\]

Equation (44) is the new identification model for the bilinear system in (4). Based on this identification model, we introduce three intermediate variables:

\[
y_y(s) := y(s) - \phi_y^T(s)\theta_y - \psi^T(s)\rho, \tag{45}
\]

\[
y_u(s) := y(s) - \phi_u^T(s)\theta_y - \psi^T(s)\rho, \tag{46}
\]

\[
y_v(s) := y(s) - \phi_y^T(s)y + \phi_u^T(s)\theta_u. \tag{47}
\]

According to the hierarchical identification principle, and from (44) we can obtain the following three subsystems:

\[
y_y(s) = \phi_y^T(s)\theta_y + v(s), \tag{48}
\]

\[
y_u(s) = \phi_u^T(s)\theta_u + v(s), \tag{49}
\]

\[
y_v(s) = \psi^T(s)\rho + v(s). \tag{50}
\]

Based on the maximum-likelihood principle in Section 3.1, we define the following cost functions

\[
J_y(\theta_y)|\theta_{y,\text{ML}} := \frac{1}{2} \sqrt{\sum_{j=(s-1)+1}^{sl} u^2(j)} |_{\theta_{y,\text{ML}}},
\]

\[
J_u(\theta_u)|\theta_{u,\text{ML}} := \frac{1}{2} \sqrt{\sum_{j=(s-1)+1}^{sl} u^2(j)} |_{\theta_{u,\text{ML}}},
\]

\[
J_v(\rho)|\rho_{\text{ML}} := \frac{1}{2} \sqrt{\sum_{j=(s-1)+1}^{sl} u^2(j)} |_{\rho_{\text{ML}}},
\]

where \( v(s) \) is given by (16).

Let \( \hat{\theta}_k(s) = [\hat{\theta}_{y,k-1}(s), \hat{\theta}_{u,k-1}(s), \hat{\rho}_{k-1}(s)]^T \) be the estimates of \( \theta(s) = [\theta_y^T(s), \theta_u^T(s), \rho^T(s)]^T \) at iteration \( k \). Minimizing \( J_y(\theta_y), J_u(\theta_u) \) and \( J_v(\rho) \), and using the negative gradient search yield the following gradient-based iterative relation for computing the parameter estimates \( \hat{\theta}_{y,k}(s), \hat{\theta}_{u,k}(s) \) and \( \hat{\rho}_{k}(s) \):

\[
\hat{\theta}_{y,k}(s) := \hat{\theta}_{y,k-1}(s) - \mu_{y,k}(s)\text{grad} \left[J_y \left(\hat{\theta}_{y,k-1}(s)\right)\right]
\]

\[
= \hat{\theta}_{y,k-1}(s) - \mu_{y,k}(s) \sum_{j=(s-1)+1}^{sl} \hat{u}_k(j) \frac{\partial v(j)}{\partial \theta_y} |_{\theta_{y,k-1}(s)} , \tag{51}
\]

\[
\hat{\theta}_{u,k}(s) := \hat{\theta}_{u,k-1}(s) - \mu_{u,k}(s)\text{grad} \left[J_u \left(\hat{\theta}_{u,k-1}(s)\right)\right]
\]

\[
= \hat{\theta}_{u,k-1}(s) - \mu_{u,k}(s) \sum_{j=(s-1)+1}^{sl} \hat{u}_k(j) \frac{\partial v(j)}{\partial \theta_u} |_{\theta_{u,k-1}(s)} , \tag{52}
\]

\[
\hat{\rho}_{k}(s) := \hat{\rho}_{k-1}(s) - \mu_{v,k}(s)\text{grad} \left[J_v \left(\hat{\rho}_{k-1}(s)\right)\right]
\]

\[
= \hat{\rho}_{k-1}(s) - \mu_{v,k}(s) \sum_{j=(s-1)+1}^{sl} \hat{u}_k(j) \frac{\partial v(j)}{\partial \rho} |_{\rho_{k-1}(s)} , \tag{53}
\]

where \( \mu_{y,k}(s), \mu_{u,k}(s) \) and \( \mu_{v,k}(s) \) are the iterative step-size. Define the information vectors \( \zeta_{y,k}(s), \zeta_{u,k}(s) \) and \( \zeta_{v,k}(s) \) as

\[
\hat{\zeta}_{y,k}(s) := - \frac{\partial v(s)}{\partial \theta_y} |_{\theta_{y,k-1}(s)}
\]

\[
= \left[\frac{\partial v(s)}{\partial a_1}, \cdots, \frac{\partial v(s)}{\partial a_n}, \frac{\partial v(s)}{\partial b_1}, \cdots, \frac{\partial v(s)}{\partial b_n}\right]^T |_{\theta_{y,k-1}(s)} \in \mathbb{R}^{2n}.
\]
\[
\tilde{\zeta}_{u,k}(s) := -\frac{\partial v(s)}{\partial \theta_u} \bigg|_{\theta_{u,k-1}(s)} \\
= -\left[ \frac{\partial v(s)}{\partial c_1}, \ldots, \frac{\partial v(s)}{\partial c_m}, \frac{\partial v(s)}{\partial d_2}, \ldots, \frac{\partial v(s)}{\partial d_n} \right]^T \in \mathbb{R}^{2n-1}, \\
\tilde{\zeta}_{v,k}(s) := -\frac{\partial v(s)}{\partial \rho} \tilde{\rho}_{k-1}(s) \\
= -\left[ \frac{\partial v(s)}{\partial c_1}, \ldots, \frac{\partial v(s)}{\partial c_m}, \frac{\partial v(s)}{\partial f_1}, \ldots, \frac{\partial v(s)}{\partial f_n} \right]^T \tilde{\rho}_{k-1}(s) \in \mathbb{R}^{n_s+n_f}.
\]

The partial derivatives of \(v(s)\) in (16) with respect to \(a_j, b_j, c_j, d_j, e_j\) and \(f_j\) at the point \(\tilde{\theta}_{k-1}(s) = [\tilde{\theta}_y^T, \tilde{\theta}_{u,k-1}(s), \tilde{\rho}_{k-1}^T(s)]^T\) can be computed by (22)–(31). Thus, we have

\[
\begin{align*}
\tilde{\zeta}_{y,k}(s) &= [-\tilde{y}_{f,k}(s-1), -\tilde{y}_{f,k}(s-2), \ldots, -\tilde{y}_{f,k}(s-n), -u(s-n)\tilde{y}_{f,k}(s-1), \\
& \quad \cdots, -u(s-n)\tilde{y}_{f,k}(s-n)]^T, \\
\tilde{\zeta}_{u,k}(s) &= [\tilde{u}_{f,k}(s-1), \tilde{u}_{f,k}(s-2), \ldots, \tilde{u}_{f,k}(s-n), u(s-n)\tilde{u}_{f,k}(s-2), \\
& \quad \cdots, u(s-n)\tilde{u}_{f,k}(s-n)]^T, \\
\tilde{\zeta}_{v,k}(s) &= [-\tilde{\w}_{f,k}(s-1), -\tilde{\w}_{f,k}(s-2), \ldots, -\tilde{\w}_{f,k}(s-n_c), \\
& \quad \tilde{\v}_{f,k}(s-1), \tilde{\v}_{f,k}(s-2), \ldots, \tilde{\v}_{f,k}(s-n_f)]^T.
\end{align*}
\]

Then Equations (51)–(53) can be written as

\[
\begin{align*}
\tilde{\theta}_{y,k}(s) &= \tilde{\theta}_{y,k-1}(s) + \mu_{y,k}(s) \sum_{j=(s-1)l+1}^{sl} \tilde{v}_k(j) \tilde{\zeta}_{y,k}(s), \\
\tilde{\theta}_{u,k}(s) &= \tilde{\theta}_{u,k-1}(s) + \mu_{u,k}(s) \sum_{j=(s-1)l+1}^{sl} \tilde{v}_k(j) \tilde{\zeta}_{u,k}(s), \\
\tilde{\rho}_{k}(s) &= \tilde{\rho}_{k-1}(s) + \mu_{v,k}(s) \sum_{j=(s-1)l+1}^{sl} \tilde{v}_k(j) \tilde{\zeta}_{v,k}(s).
\end{align*}
\]

Define the estimate of \(\psi(s)\) as

\[
\hat{\psi}_k := [-\hat{\w}_{k-1}(s-1), -\hat{\w}_{k-1}(s-2), \ldots, -\hat{\w}_{k-1}(s-n_c), \hat{\v}_{k-1}(s-1), \\
\hat{\v}_{k-1}(s-2), \cdots, \hat{\v}_{k-1}(s-n_f)]^T \in \mathbb{R}^{n_s+n_f}.
\]

From (44), we have \(v(s-j) = y(s-j) - \phi_y^T(s-j)\theta_y - \phi_u^T(s-j)\theta_u - \psi^T(s-j)\rho\). Replacing \(\psi(s)\), \(\theta_y\), \(\theta_u\) and \(\rho\) with their estimates \(\hat{\psi}_k(s), \hat{\theta}_{y,k}(s), \hat{\theta}_{u,k}(s)\) and \(\hat{\rho}_{k}(s)\), the estimate of \(v(s)\) can be computed through

\[
\hat{v}_k(s-j) = y(s-j) - \phi_y^T(s-j)\hat{\theta}_y(s) - \phi_u^T(s-j)\hat{\theta}_u(s) - \hat{\psi}_k^T(s-j)\hat{\rho}_k(s).
\]

From (43), we have \(w(s-j) = y(s-j) - \phi_y^T(s-j)\theta_y - \phi_u^T(s-j)\theta_u\). Replacing \(\theta_y\) and \(\theta_u\) with their estimates \(\hat{\theta}_{y,k}(s)\) and \(\hat{\theta}_{u,k}(s)\) gives the estimate of \(w(s)\):

\[
\hat{w}_k(s-j) = y(s-j) - \phi_y^T(s-j)\hat{\theta}_y(s) - \phi_u^T(s-j)\hat{\theta}_u(s).
\]

Using the above definitions and (61), and from (57)–(59), we have

\[
\begin{align*}
\hat{\theta}_{y,k}(s) &= \hat{\theta}_{y,k-1}(s) + \mu_{y,k}(s) \sum_{j=(s-1)l+1}^{sl} \hat{\zeta}_{y,k}(s) \\
& \quad \times [y(j) - \phi_y^T(j)\hat{\theta}_{y,k-1}(s) - \phi_u^T(j)\hat{\theta}_{u,k-1}(s) - \hat{\psi}_k^T(j)\hat{\rho}_{k-1}(s)], \\
\hat{\theta}_{u,k}(s) &= \hat{\theta}_{u,k-1}(s) + \mu_{u,k}(s) \sum_{j=(s-1)l+1}^{sl} \hat{\zeta}_{u,k}(s) \\
& \quad \times [y(j) - \phi_y^T(j)\hat{\theta}_{y,k}(s) - \phi_u^T(j)\hat{\theta}_{u,k}(s) - \hat{\psi}_k^T(j)\hat{\rho}_{k-1}(s)],
\end{align*}
\]
\[
\hat{\rho}_k(s) = \hat{\rho}_{k-1}(s) + \mu_{v,k}(s) \sum_{j=(s-1)l+1}^{sl} \hat{\zeta}_{v,k}(s) \\
\times [y(j) - \hat{\theta}_y^T(j)\hat{\theta}_y(k) - \hat{\theta}_u^T(j)\hat{\theta}_u(k) - \hat{\psi}_k(j)\hat{\rho}_{k-1}(s)].
\]

In order to guarantee the convergence of \(\hat{\theta}_{y,k}(s)\), \(\hat{\theta}_{u,k}(s)\) and \(\hat{\rho}_k(s)\), the conservative choice of \(\mu_{y,k}(s)\), \(\mu_{u,k}(s)\) and \(\mu_{v,k}(s)\) are to satisfy

\[
0 < \mu_{y,k}(s) \leq 2\lambda_{max}^{-1} \sum_{j=(s-1)l+1}^{sl} \hat{\zeta}_{y,k}(j)\hat{\zeta}_{y,k}^T(j),
\]

\[
0 < \mu_{u,k}(s) \leq 2\lambda_{max}^{-1} \sum_{j=(s-1)l+1}^{sl} \hat{\zeta}_{u,k}(j)\hat{\zeta}_{u,k}^T(j),
\]

\[
0 < \mu_{v,k}(s) \leq 2\lambda_{max}^{-1} \sum_{j=(s-1)l+1}^{sl} \hat{\zeta}_{v,k}(j)\hat{\zeta}_{v,k}^T(j).
\]

Equations (28)–(31), (41)–(42), (54)–(56) and (60)–(68) form the hierarchical maximum-likelihood gradient-based iterative (H-ML-GI) identification algorithm for the bilinear systems.

The identification steps of the H-ML-GI algorithm to compute \(\hat{\theta}_k(s) = [\hat{\theta}_y^T(s), \hat{\theta}_u^T(s), \hat{\rho}_k^T(s)]^T\) are listed as follows.

1. Let \(s = 1\) and set \(\hat{\theta}_{y,0}(s) = 1_{2n/p_0}, \hat{\theta}_{u,0}(s) = 1_{2n-1/p_0}\) and \(\hat{\rho}_0(s) = 1_{n_e+n_f/p_0}, p_0 = 10^6\), give the data length \(l\), a small \(\varepsilon > 0\) and a maximum iteration \(k_{max}\).
2. Collect the input-output data \(\{u(i), y(i): i = (s-1)l+1, (s-1)l+2, \cdots, sl\}\).
3. Let \(k = 1, \) and set the initial values: \(\hat{v}_0(s-j)\) and \(\hat{v}_0(s-j)\) are random numbers, \(j = 1, 2, \cdots, \max[n_e, n_f]\). Set \(\hat{y}_{f,0}(s-i), \hat{w}_{f,0}(s-i)\) and \(\hat{w}_{f,0}(s-i)\) be random numbers, \(i = 1, 2, \cdots, \max[n, n_e, n_f]\).
4. Form \(\hat{\phi}_y(s)\) and \(\hat{\phi}_u(s)\) using (41) and (42), and form \(\hat{\psi}_k(s)\) using (60).
5. Compute \(\hat{y}_{f,k}(s)\) using (28), \(\hat{w}_{f,k}(s)\) using (29) and \(\hat{v}_{f,k}(s)\) using (31).
6. Form \(\hat{\zeta}_{y,k}(s)\) using (54), \(\hat{\zeta}_{u,k}(s)\) using (55), and \(\hat{\zeta}_{v,k}(s)\) using (56).
7. Select a large \(\mu_{y,k}(s)\) satisfying (66), a large \(\mu_{u,k}(s)\) satisfying (67) and a large \(\mu_{v,k}(s)\) satisfying (68), and update the estimates \(\hat{\theta}_{y,k}(s)\) using (63), \(\hat{\theta}_{u,k}(s)\) using (64) and \(\hat{\rho}_k(s)\) using (65).
8. Compute \(\hat{\phi}_k(s-j)\) using (62), \(\hat{v}_{k}(s-j)\) using (61).
9. If \(k < k_{max}\), increase \(k\) by 1 and go to Step 4; otherwise, proceed to next step.
10. Compare \(\hat{\theta}_k(s) = [\hat{\theta}_y^T(s), \hat{\theta}_u^T(s), \hat{\rho}_k^T(s)]^T\) with \(\vartheta(s) = [\theta_y^T(s), \theta_u^T(s), \rho(s)]^T\), if \(||\hat{\theta}_k(s) - \hat{\theta}_k(s)| > \varepsilon\) set \(\hat{\theta}_{0}(s+1) = \hat{\theta}_k(s)\), increase \(s\) by 1 and go to Step 2; otherwise, obtain the parameter estimate \(\hat{\theta}_k(s)\) and terminate the calculation process.

The flowchart of computing \(\hat{\theta}_k(s)\) in the H-ML-GI algorithm is shown in Figure 3.

### 5 The Gradient-based iterative algorithm

To show the advantages of the proposed algorithms, we give the gradient-based iterative algorithm for identifying the parameter vector \(\vartheta\) of bilinear systems.

Consider the newest \(l\) data from \((s-1)l+1\) to \(sl\) \((l \gg 4n + n_e + n_f - 1)\), and define the stacked output vector \(Y(l, s)\) and the stacked information matrix \(\Phi(l, s)\) as

\[
Y(l, s) := [y(sl), y((sl-1)+1), \cdots, y((s-1)l+1)]^T \in \mathbb{R}^l, \quad (69)
\]

\[
\Phi(l, s) := [\varphi(sl), \varphi((sl-1)+1), \cdots, \varphi((s-1)l+1)]^T \in \mathbb{R}^{l \times (4n + n_e + n_f - 1)}. \quad (70)
\]

Note that \(Y(l, s)\) and \(\Phi(l, s)\) contain all the measured data \(\{u(i), y(i): i = 1, 2, \cdots, s\}\). Let \(\|X\|^2 = tr[XX^T]\) and define a quadratic criterion function:

\[
J_l(\vartheta) := ||Y(l, s) - \Phi(l, s)\vartheta||^2.
\]
Minimizing $J_1(\theta)$ and using the negative gradient search lead to the iterative relation for computing the parameter estimates $\hat{\theta}_k(s)$:

$$\hat{\theta}_k(s) = \hat{\theta}_{k-1}(s) + \mu_{1,k}(s)\Phi^T(l,s)[Y(l,s) - \Phi(l,s)\hat{\theta}_{k-1}(s)],$$

(71)

where $\mu_{1,k}(s)$ is an iterative step-size or a convergence factor. Notice that the information vector $\varphi(s)$ in $\Phi(l,s)$ contains the unmeasured noise terms $w(s-j)$ ($j = 1,2,\cdots,n_e$) and $v(s-j)$ ($j = 1,2,\cdots,n_f$), so Equation (71) cannot give the estimate $\hat{\theta}_k(s)$ directly. The solution is to replace the unknown noise terms $w(s-j)$ and $v(s-j)$ in $\varphi(s)$ with their corresponding estimates $\hat{w}_{k-1}(s-j)$ and $\hat{v}_{k-1}(s-j)$ at iteration $k-1$ and to define the estimate of $\varphi(s)$ as

$$\hat{\varphi}_k(s) := [\Phi^T(s), \hat{\psi}_k(s)]^T,$$

(72)

$$\hat{\psi}_k(s) := [-\hat{w}_{k-1}(s-1), -\hat{w}_{k-1}(s-2), \cdots, -\hat{w}_{k-1}(s-n_e),$$

$$\hat{v}_{k-1}(s-1), \hat{v}_{k-1}(s-2), \cdots, \hat{v}_{k-1}(s-n_f)]^T,$$

(73)

$$\hat{\Phi}_k(l,s) := [\hat{\varphi}_k(sl), \hat{\varphi}_k(sl-1), \cdots, \hat{\varphi}_k((s-1)l + 1)]^T.$$

(74)
According to the identification model in (13), we have \( v(s - j) = y(s - j) - \varphi^T(s - j)\theta \). Replacing \( \varphi(s) \) and \( \theta \) with \( \hat{\varphi}_k(s) \) and \( \hat{\theta}_k(s) \), we can obtain the estimate of \( v(s) \) as

\[
\hat{v}_k(s - j) = y(s - j) - \hat{\varphi}_k^T(s - j)\hat{\theta}_k(s). \tag{75}
\]

From (11), we have \( w(s - j) = y(s - j) - \Phi^T(s - j)\theta \). Replacing \( \theta \) with \( \hat{\theta}_k(s) \) gives the estimate of \( w(s) \):

\[
\hat{w}_k(s - j) = y(s - j) - \Phi^T(s - j)\hat{\theta}_k(s). \tag{76}
\]

Replacing \( \Phi(l, s) \) in (71) with its estimate \( \hat{\Phi}_k(l, s) \), we can obtain the gradient-based iterative identification algorithm for estimating \( \theta \):

\[
\hat{\theta}_k(s) = \hat{\theta}_{k-1}(s) + \mu_{1,k}(s)\hat{\Phi}_k^T(l, s)[Y(l, s) - \hat{\Phi}_k(l, s)\hat{\theta}_{k-1}(s)]. \tag{77}
\]

In order to guarantee the convergence of \( \hat{\theta}_k(s) \), one conservative choice of \( \mu_{1,k}(s) \) is to satisfy

\[
0 < \mu_{1,k}(s) \leq \frac{2}{\lambda_{\max}[\hat{\Phi}_k^T(l, s)\hat{\Phi}_k(l, s)]}, \tag{78}
\]

where \( \lambda_{\max}[X] \) denotes the maximum eigenvalue of the square matrix \( X \).

Equations (7), (17)–(19), (69) and (72)–(78) form the gradient-based iterative (GI) identification algorithm for the bilinear systems.

The identification steps of the GI algorithm to compute \( \hat{\theta}_k(s) \) are listed as follows.

1. Let \( s = 1 \), and set the data length \( l \) and \( \hat{\theta}_0(s) = 1_{4n + n_e + n_f - 1/p_0} \). \( p_0 = 10^6 \). Collect the data \( \{u(j), y(j)\}: j = 0, 1, \cdots, l - 1 \}, \) and give a small positive number \( \varepsilon \) and a maximum iteration \( k_{\text{max}} \).

2. Collect the input-output data \( u(s) \) and \( y(s) \) and form the stacked output vector \( Y(l, s) \) using (69).

3. Let \( k = 1 \), and set the initial values: \( \hat{w}_0(s - j) \) and \( \hat{v}_0(s - j) \) are random numbers \( (j = 1, 2, \cdots, \max[n_e, n_f]) \).

4. Form \( \phi(s) \) using (7), form \( \psi_k(s) \) using (73), form \( \hat{\varphi}_k(s) \) using (72), and form \( \hat{\Phi}_k(l, s) \) using (74).

5. Select a large \( \mu_{1,k}(s) \) satisfying (78), and update the estimate \( \hat{\theta}_k(s) \) using (77).

6. Read \( \hat{\theta}_k(s) \) from \( \hat{\theta}_k(s) \) in (17), and compute \( \hat{w}_k(s - j) \) and \( \hat{v}_k(s - j) \) using (76) and (75), respectively.

7. If \( k < k_{\text{max}} \), increase \( k \) by 1 and go to Step 4; otherwise, proceed to next step.

8. Compare \( \|\hat{\theta}_k(s) - \hat{\theta}_{k-1}(s)\| > \varepsilon \), set \( \hat{\theta}_0(s + 1) = \hat{\theta}_k(s) \), increase \( s \) by 1 and go to Step 2; otherwise, obtain the parameter estimate \( \hat{\theta}_k(s) \) and terminate the calculation process.

The flowchart of computing \( \hat{\theta}_k(s) \) in the GI algorithm is shown in Figure 4.

### 6 Comparison of the computational efficiency

The computational efficiency of the algorithm can be evaluated by the number of multiplications (including divisions) and additions (including subtractions). One multiplication or one addition is called a flop, which means the floating point operations. The sum of the flops is the computational efficiency of the algorithm. The computational efficiency of the ML-GI algorithm, the H-ML-GI algorithm and the GI algorithm is shown in Tables 1–3, and we define \( n_0 = 4n + n_e + n_f - 1 \).

In order to make it clear, an example is taken for illustration. Taking the system orders \( n = 5, n_e = 5, n_f = 5 \) and the data length \( l = 3000 \), we have \( N_1 = 6918870, N_2 = 2970310, N_3 = 5682870, \) that is, the difference between the total flops of the ML-GI algorithm and the H-ML-GI algorithm at each iteration is \( N_1 - N_2 = 6918870 - 2970310 = 3948560 \), and the difference between the total flops of the GI algorithm and the H-ML-GI algorithm at each iteration is \( N_3 - N_2 = 5682870 - 2970310 = 2712560 \). The algorithm with higher computational efficiency means the algorithm has fewer flops. Obviously, the H-ML-GI algorithm have a higher computational efficiency than the ML-GI algorithm and the GI algorithm.
7 Example

Consider the following bilinear system:

\[ [A(r^{-1}) + u(s - n)B(r^{-1})]y(s) = [C(r^{-1}) + u(s - n)D(r^{-1})]u(s) + w(s), \]

where \( A, B, C, D, r, u, n, w \) are matrices, vectors, and constants, respectively.
Highly efficient maximum-likelihood identification methods for bilinear systems with colored noises

\[ \hat{y}(s) = y(s) - \Phi^T(s) \hat{\theta}(s) \]

**Table 2** The computational efficiency of the H-ML-GI algorithm

| Expressions | Number of multiplications | Number of additions |
|-------------|---------------------------|---------------------|
| \( \theta_{y,k}(s) = \theta_{y,k-1}(s) + \mu_{y,k}(s) \) | \( 2n(l+1) \) | \( 2nl \) |
| \( \alpha_1^{k,j} := y(j) - \Phi^T(j) \hat{\theta}_{y,k-1}(s) \) | \( n_0l \) | \( n_0l \) |
| \( \mu_{y,k}(s) = 2\lambda_{\max} \left[ \sum_{j=-r}^{l} \hat{\zeta}_{y,k}(j) \right] \) | \( (2n)^2(l+1) + 1 \) | \( (2n)^2l - 1 \) |
| \( \hat{\theta}_{u,k}(s) = \hat{\theta}_{u,k-1}(s) + \mu_{u,k}(s) \) | \( 2n(l-1) \) | \( 2nl \) |
| \( \alpha_2^{k,j} := y(j) - \Phi^T(j) \hat{\Phi}_{y,k}(s) \) | \( n_0l \) | \( n_0l \) |
| \( \mu_{u,k}(s) = 2\lambda_{\max} \left[ \sum_{j=-r}^{l} \hat{\zeta}_{u,k}(j) \right] \) | \( (2n)^2(l+1) + 1 \) | \( (2n)^2l - 1 \) |
| \( \hat{\mu}_{u,k}(s) = \hat{\mu}_{u,k-1}(s) + \mu_{u,k}(s) \) | \( (n_e + n_f)(l+1) \) | \( (n_e + n_f)l \) |
| \( \alpha_3^{k,j} := y(j) - \Phi^T(j) \hat{\theta}_{y,k}(s) \) | \( n_0l \) | \( n_0l \) |
| \( \mu_{y,k}(s) = 2\lambda_{\max} \left[ \sum_{j=-r}^{l} \hat{\zeta}_{v,k}(j) \right] \) | \( (n_e + n_f)^2(l+1) + 1 \) | \( (n_e + n_f)^2l - 1 \) |
| \( \hat{y}_f(s) \) | \( (n_e + n_f)l \) | \( (n_e + n_f)l \) |
| \( \hat{\mu}_f(s) \) | \( (n_e + n_f)l \) | \( (n_e + n_f)l \) |
| \( \hat{w}_f(s) \) | \( (4n + 1 + n_f)l \) | \( (4n - 1 + n_f)l \) |
| \( \hat{v}_f(s) = y(s) - \Phi^T(s) \hat{\theta}_k(s) \) | \( n_0l \) | \( n_0l \) |
| \( \hat{w}_f(s) = y(s) - \Phi^T(s) \hat{\theta}_k(s) \) | \( (4n - 4)l \) | \( (4n - 1)l \) |
| **Sum** | \( [(2n)^2 + (2n - 1)^2 + (n_e + n_f)^2] \) | \( [(2n)^2 + (2n - 1)^2] \) |
| | \(+7n_0 + 2n_f + 2)n_l + n_0 + 3 \) | \(+n_e + n_f)^2 + 7n_0 \) |
| | \(+2n^2 + (2n - 1)^2 + (n_e + n_f)^2 + 2n_f(\text{l-3}) \) | \(+2n^2 + 2(n - 1)^2 + (n_e + n_f)^2 + 14n_0 + 4n_f + 2\text{l}\) |
| **Total flops** | \( N_2 = [8n^2 + 2(2n - 1)^2 + 2(n_e + n_f)^2 + 14n_0 + 4n_f + 2]\text{l} \) | \( +2n^2 + 2(n - 1)^2 + (n_e + n_f)^2 + n_0 \) |

**Table 3** The computational efficiency of the GI algorithm

| Expressions | Number of multiplications | Number of additions |
|-------------|---------------------------|---------------------|
| \( \hat{\theta}_k(s) = \hat{\theta}_{k-1}(s) + \mu_{1,k}(s) \Phi^T(l,s) \alpha_k \in \mathbb{R}^{n_0} \) | \( n_0(l+1) \) | \( n_0l \) |
| \( \alpha_k := Y(l,s) - \Phi(l,s) \hat{\theta}_{k-1}(s) \in \mathbb{R}^l \) | \( n_0l \) | \( n_0l \) |
| \( \mu_{1,k}(s) = 2\lambda_{\max} \left[ \Phi^T(l,s) \right] \) | \( n_0^2l + 4n_0 + 4n - 1 \) | \( n_0^2l - 1 \) |
| \( \hat{v}_k(s) = y(s) - \Phi^T(s) \hat{\theta}_k(s) \in \mathbb{R}^l \) | \( n_0^3l + 3n_0 + 4n - 1 \) | \( n_0^3l - 1 \) |
| \( \hat{w}_k(s) = y(s) - \Phi^T(s) \hat{\theta}_k(s) \in \mathbb{R}^l \) | \( (4n - 1)l \) | \( (4n - 1)l \) |
| **Sum** | \( (n_0^3 + 3n_0 + 4n - 1)l \) | \( (n_0^3 + 3n_0 + 4n - 1)l - 1 \) |
| | \(+n_0^3 + n_0 + 1 \) | \(+n_0^3 + n_0 + 1 \) |
| **Total flops** | \( N_3 = (2n_0^2 + 6n_0 + 8n - 2)l + n_0^2 + n_0 \) | \(| \) |

\[ E(r^{-1})w(s) = F(r^{-1})v(s), \]
\[ A(r^{-1}) = 1 + a_1r^{-1} + a_2r^{-2} = 1 + 0.52r^{-1} + 0.76r^{-2}, \]
\[ B(r^{-1}) = b_1r^{-1} + b_2r^{-2} = 0.1r^{-1} + 0.21r^{-2}, \]
\[ C(r^{-1}) = c_1r^{-1} + c_2r^{-2} = 1.2r^{-1} - 2.19r^{-2}, \]
\[ D(r^{-1}) = d_2r^{-2} = 0.12r^{-2}, \]
\[ E(r^{-1}) = 1 + e_1r^{-1} = 1 - 0.05r^{-1}, \]
\[ F(r^{-1}) = 1 + f_1 r^{-1} = 1 + 0.1 r^{-1}. \]

The parameter vector to be estimated is given by
\[
\vartheta = [\theta^T(s), \theta^T_u(s), \rho^T(s)]^T = [a_1, a_2, b_1, b_2, c_1, c_2, d_2, e_1, f_1]^T = [0.52, 0.76, 0.1, -0.21, 1.2, -2.19, 0.12, -0.05, 0.1]^T.
\]

In simulation, the input \( \{u(s)\} \) is taken as a persistent excitation sequence with zero mean and unit variance, and is shown in Figure 5. \( \{v(s)\} \) is an uncorrelated noise sequence with zero mean and variance \( \sigma^2 \).

**Case I: About the parameter estimation**

Taking the noise variance \( \sigma^2 = 1.00^2 \) and \( \sigma^2 = 2.00^2 \), respectively, taking the data length \( l = 3000 \), applying the ML-GI algorithm, the H-ML-GI algorithm and the GI algorithm to estimate the parameters of the system, the parameter estimates and their errors with different noise variances are shown in Tables 4–5, and the parameter estimates and the estimation errors \( \delta \) versus \( k \) are plotted in Figures 6–10, where \( \delta := \|\hat{\vartheta}_k(s) - \vartheta\|/\|\vartheta\| \).

![Fig. 5 The input u(s) of the system](image)

![Fig. 6 The ML-GI estimates versus k with \( \sigma^2 = 1.00^2 \)](image)

From the simulation results in Tables 4–5 and Figures 6–10, we can draw the following conclusions.

– The parameter estimation errors of the ML-GI, H-ML-GI and GI algorithms become smaller as the iteration \( k \) increase – see Tables 4–5 and Figures 6–8.
Highly efficient maximum-likelihood identification methods for bilinear systems with colored noises

Fig. 7 The H-ML-GI estimates versus $k$ with $\sigma^2 = 1.00^2$

Fig. 8 The GI estimates versus $k$ with $\sigma^2 = 1.00^2$

Fig. 9 The estimation errors $\delta$ versus $k$ with different algorithms ($\sigma^2 = 1.00^2$)
Table 4  The estimates and errors with different algorithms ($\sigma^2 = 1.00^2$)

| Algorithms | $k$ | $a_1$ | $a_2$ | $b_1$ | $b_2$ | $c_1$ | $c_2$ | $d_1$ | $d_2$ | $e_1$ | $e_2$ | $f_1$ | $\delta$ (%) |
|------------|----|------|------|------|------|------|------|------|------|------|------|------|------|
| ML-GI      | 1  | 0.15863 | 0.25508 | 0.06806 | -0.11431 | 0.01852 | -0.04752 | -0.00046 | 0.15863 | -0.00083 | 94.87897 |
|            | 2  | 0.25800 | 0.42124 | 0.07756 | -0.16152 | 0.03101 | -0.08819 | 0.00124 | 0.21702 | -0.08169 | 92.18379 |
|            | 5  | 0.43535 | 0.72239 | 0.07619 | -0.20444 | 0.04996 | -0.14213 | 0.03518 | 0.26213 | -0.17605 | 85.18774 |
|            | 10 | 0.75100 | 1.30519 | 0.08436 | -0.20817 | 0.06349 | -0.25238 | 0.03765 | 0.30755 | -0.18630 | 78.94075 |
|            | 50 | 0.53061 | 0.76140 | 0.09929 | -0.20781 | 0.10860 | -0.21801 | 0.06388 | 0.08062 | -0.06750 | 26.90475 |
|            | 100| 0.52435 | 0.75899 | 0.10214 | -0.20780 | 0.10799 | -0.07076 | 0.08191 | 0.11574 |
| H-ML-GI    | 1  | 0.24475 | 0.39356 | 0.10501 | -0.17637 | 0.63885 | -1.41113 | 0.07596 | 0.17683 | -0.00072 | 40.86042 |
|            | 2  | 0.36185 | 0.59458 | 0.09604 | -0.19840 | 0.85020 | -1.88609 | 0.08592 | 0.18612 | -0.03529 | 21.85280 |
|            | 5  | 0.49535 | 0.73269 | 0.10014 | -0.20879 | 1.15691 | -2.19649 | 0.10277 | 0.01131 | 0.11389 | 3.23587 |
|            | 10 | 0.52179 | 0.75746 | 0.10212 | -0.20788 | 1.20454 | -2.20300 | 0.10785 | 0.01231 | 0.13775 | 2.11392 |
|            | 20 | 0.52440 | 0.75960 | 0.10211 | -0.20780 | 1.20662 | -2.19676 | 0.10779 | 0.02736 | 0.14650 |
|            | 50 | 0.52442 | 0.75965 | 0.10211 | -0.20780 | 1.20667 | -2.19673 | 0.10776 | 0.03679 | 0.15847 |
|            | 100| 0.52442 | 0.75970 | 0.10212 | -0.20780 | 1.20664 | -2.19673 | 0.10770 | 0.11387 | 0.98416 |
|            | 200| 0.52443 | 0.75979 | 0.10213 | -0.20780 | 1.20671 | -2.19683 | 0.10779 | 0.04781 | 0.11587 |
|            | 500| 0.52439 | 0.75979 | 0.10214 | -0.20780 | 1.20671 | -2.19683 | 0.10778 | 0.13426 |
| GI         | 1  | 0.24475 | 0.39356 | 0.10501 | -0.17637 | 0.63885 | -1.41113 | 0.07596 | 0.17683 | -0.00072 | 40.86042 |
|            | 2  | 0.36185 | 0.59458 | 0.09604 | -0.19840 | 0.85020 | -1.88609 | 0.08592 | 0.18612 | -0.03529 | 21.85280 |
|            | 5  | 0.49535 | 0.73269 | 0.10014 | -0.20879 | 1.15691 | -2.19649 | 0.10277 | 0.01131 | 0.11389 | 3.23587 |
|            | 10 | 0.52179 | 0.75746 | 0.10212 | -0.20788 | 1.20454 | -2.20300 | 0.10785 | 0.01231 | 0.13775 | 2.11392 |
|            | 20 | 0.52440 | 0.75960 | 0.10211 | -0.20780 | 1.20662 | -2.19676 | 0.10779 | 0.02736 | 0.14650 |
|            | 50 | 0.52442 | 0.75965 | 0.10211 | -0.20780 | 1.20667 | -2.19673 | 0.10776 | 0.03679 | 0.15847 |
|            | 100| 0.52442 | 0.75970 | 0.10212 | -0.20780 | 1.20664 | -2.19673 | 0.10770 | 0.11387 | 0.98416 |
|            | 200| 0.52443 | 0.75979 | 0.10213 | -0.20780 | 1.20671 | -2.19683 | 0.10779 | 0.04781 | 0.11587 |
|            | 500| 0.52439 | 0.75979 | 0.10214 | -0.20780 | 1.20671 | -2.19683 | 0.10778 | 0.13426 |

True values 0.52000 0.76000 0.10000 -0.21000 1.20000 -2.19000 0.12000 -0.05000 0.10000

Fig. 10  The estimation errors $\delta$ versus $k$ with different algorithms ($\sigma^2 = 2.00^2$)

- Under the same data length and noise variances, the ML-GI algorithm and the H-ML-GI algorithm have smaller estimation errors than the GI algorithm – see Tables 4 and 5. In other words, the parameter estimates accuracy of the ML-GI algorithm and the H-ML-GI algorithm are higher than those of the GI algorithm.
- Among these three algorithms, the convergence rate of the H-ML-GI algorithm is obviously faster than that of the ML-GI algorithm and the GI algorithm – see Figures 9–10.
- In this example, the system orders are $n = 2$, $n_e = 1$, $n_f = 1$, and the data length is $l = 3000$, and the number of iterations is $k = 500$. According to Tables 1–3, the total flops of the ML-GI algorithm, the H-ML-GI algorithm and the GI algorithm are $kN_1 = 500 \times 1062090 = 1,000,895,000$. Under the same data length and noise variances, the ML-GI algorithm and the H-ML-GI algorithm have smaller estimation errors than the GI algorithm – see Tables 4 and 5. In other words, the parameter estimates accuracy of the ML-GI algorithm and the H-ML-GI algorithm are higher than those of the GI algorithm.
mean-squares errors (MSEs) and define MSE as

\[ E(\hat{y}_i) := \frac{1}{l} \sum_{j=l+1}^{l+l_r} [\hat{y}_i(s) - y(s)]^2 = 0.99769, \]

\[ E(\hat{y}_2) := \frac{1}{l} \sum_{j=l+1}^{l+l_r} [\hat{y}_2(s) - y(s)]^2 = 0.99955, \]

\[ E(\hat{y}_3) := \frac{1}{l} \sum_{j=l+1}^{l+l_r} [\hat{y}_3(s) - y(s)]^2 = 0.99632. \]

From Figure 11, we can see that the predicted outputs \( \hat{y}_1(s) \), \( \hat{y}_2(s) \) and \( \hat{y}_3(s) \) are very close to the true output \( y(s) \), and the MSEs of the ML-GI algorithm, the H-ML-GI algorithm and the GI algorithm are very close to the noise variance \( \sigma^2 = 1.00^2 \). In other words, the estimated models can capture the dynamics of the system.
8 Conclusions

This work mainly considers the iterative identification problems of a special class of nonlinear systems with unmeasured disturbances. By combining the maximum-likelihood principle and the negative gradient search principle, a maximum-likelihood gradient-based iterative (ML-GI) algorithm is derived for the parameter estimation of bilinear systems with an ARMA process. Based on the hierarchical identification principle, a hierarchical maximum-likelihood gradient-based iterative (H-ML-GI) algorithm is developed for improving the computational efficiency. For comparison, a gradient-based iterative (GI) algorithm is also presented. The simulation results verify that the proposed algorithms can generate more accurate estimates for bilinear systems than the GI algorithm, and the H-ML-GI algorithm has a faster convergence rate and a higher computational efficiency than the ML-GI algorithm and the GI algorithm. The proposed algorithms can be extended to solve the parameter estimation problems of other nonlinear systems with colored noises or the nonlinear systems with interval-varying measurements, and can be applied to other fields such as the industrial process.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61472195) and the Natural Science Foundation of Shandong Province (ZR2020QF065, ZR2020MF081).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest regarding the publication of this paper.

References

1. M. Chen, S.S. Ge, Adaptive neural output feedback control of uncertain nonlinear systems with unknown hysteresis using disturbance observer. IEEE Transactions on Industrial Electronics 62(12), 7706-7716 (2015)
2. E. Garone, M.M. Nicotra, Explicit reference governor for constrained nonlinear systems. IEEE Transactions on Automatic Control 61(5), 1379-1384 (2016)
3. J. Prawin, A. Rama Mohan Rao, Damage detection in nonlinear systems using an improved describing function approach with limited instrumentation. Nonlinear Dynamics 96, 1447-1470 (2019)
4. F. Ding, J. Pan, A. Alsaeedi, et al, Gradient-based iterative parameter estimation algorithms for dynamical systems from observation data. Mathematics 7(3), (2019) Article Number: 428. http://dx.doi.org/10.3390/math7030428.
5. J.H. Li, J.L. Zhang, Maximum likelihood identification of dual-rate Hammerstein output-error moving average system. IET Control Theory and Applications 14(8), 1089-1101 (2020)
6. H. Modares, A. Alfi, M.N. Sistani, Parameter estimation of bilinear systems based on an adaptive particle swarm optimization. Engineering Applications of Artificial Intelligence 23(7), 1105-1111 (2010)
7. C.W. Wang, L. Quan, Z.X. Jiao, et al, Nonlinear adaptive control of hydraulic system with observing and compensating mismatching uncertainties. IEEE Transactions on Control Systems Technology 26(3), 927-938 (2018)
Highly efficient maximum-likelihood identification methods for bilinear systems with colored noises

8. O. Sarfraz, C.K. Bach, C.R. Bradshaw, Discrete modeling of fin-and-tube heat exchangers with cross-fin conduction functionality. International Journal of Refrigeration 104, 270-281 (2019)
9. S.H. Park, Y.H. Kim, Y.S. Kim, et al, Numerical study on the effect of different hole locations in the fan case on the thermal performance inside a gas oven range. Applied Thermal Engineering 137, 123-133 (2018)
10. J.N. Juang, Continuous-time bilinear system identification. Nonlinear Dynamics 39(1-2), 79-94 (2005)
11. Z. Hafezi, M.M. Arefi, Recursive generalized extended least squares and RML algorithms for identification of bilinear systems with ARMA noise. ISA Transactions 88, 50-61 (2019)
12. J.R. Li, Z.G. Feng, Y.X. Zhao, et al, Reachable set estimation for discrete-time bilinear systems with time-varying delays. Journal of the Franklin Institute 355(13), 5721-5735 (2018)
13. W. Favoreel, B.D. Moor, F.V. Overschee, Subspace identification of bilinear systems subject to white inputs. IEEE Transactions on Automatic Control 44(6), 1157-1165 (1999)
14. N. Berk Hizir, M.Q. Phan, R. Betti, R.W. Longman, Identification of discrete-time bilinear systems through equivalent linear models. Nonlinear Dynamics 69(4), 2065-2078 (2012)
15. M.Q. Phan, H. Celik, A superspace method for discrete-time bilinear model identification by interaction matrices. The Journal of the Astronautical Sciences 59(1), 421-440 (2012)
16. F. Vicario, M.Q. Phan, R. Betti, R.W. Longman, Linear state representations for identification of bilinear discrete-time models by interaction matrices. Nonlinear Dynamics 77(4), 1561-1576 (2014)
17. H. Dai, N.K. Sinha, Robust recursive least-squares method with modified weights for bilinear system identification. IEEE Proceedings D – Control Theory and Applications 136(3), 122-126 (1989)
18. M.H. Li, X.M. Liu, The least squares based iterative algorithms for parameter estimation of a bilinear system with autoregressive noise using the data filtering technique. Signal Processing 147, 23-34 (2018)
19. X. Zhang, F. Ding, L. Xu, Recursive parameter estimation methods and convergence analysis for a special class of nonlinear systems. International Journal of Robust and Nonlinear Control 30(4), 1373-1393 (2020)
20. D.D. Meng, Recursive least squares and multi-innovation gradient estimation algorithms for bilinear stochastic systems. Circuits Systems and Signal Processing 36(3), 1052-1065 (2017)
21. F. Fnaiech, L. Ljung, Recursive identification of bilinear systems. International Journal of Control 45(2), 453-470 (1987)
22. M.M. Gabe, T.S. Rao, On the identification of bilinear systems from operating records. Theory and Application of Digital Control 40(1), 375-380 (1982)
23. M.H. Li, X.M. Liu, Maximum likelihood least squares based iterative estimation for a class of bilinear systems using the data filtering technique. International Journal of Control Automation and Systems 18(6), 1581-1592 (2020)
24. N.Kawaria, R. Patiladar, N.V. George, Parameter estimation of MIMO bilinear systems using a Levy shuffled frog leaping algorithm. Neurocomputing 211(14), 3849-3858 (2017)
25. F.R. Shi, H.L. Li, S-X. Yang, et al, Novel maximum likelihood estimation of clock skew in one-way broadcast time synchronization. IEEE Transactions on Industrial Electronics 67(11), 9948-9957 (2020)
26. K. Kalbasi, S.J. Golestani, On the relaxed maximum-likelihood blind MIMO channel estimation for orthogonal space-time block codes. Signal Processing 175, (2020) Article Number: 107653.
27. M.T. Chen, F. Ding, R.M. Lin, et al, Maximum likelihood least squares-based iterative methods for output error bilinear-parameter models with colored noises. International Journal of Robust and Nonlinear Control 30(15), 6262-6280 (2020)
28. A. Alfonsi, A. Kebaier, C. Rey, Maximum likelihood estimation for Wishart processes. Stochastic Processes and their Applications 126(11), 3243-3282 (2016)
29. M.S. Aslam, Maximum likelihood least squares identification method for active noise control systems with autoregressive moving average noise. Automatica 69, 1-11 (2016)
30. H.F. Xia, Y. Ji, L. Xu, et al, Maximum likelihood-based gradient estimation for multivariable nonlinear systems using the multi-innovation identification theory. International Journal of Robust and Nonlinear Control 30(14), 5446-5463 (2020)
31. S. Gibson, A. Wills, B. Ninness, Maximum-likelihood parameter estimation of bilinear systems. IEEE Transactions on Automatic Control 50(10), 1581-1596 (2005)
32. J.H. Li, T.C. Zong, J.P. Gu, et al, Parameter estimation of Wiener systems based on the particle swarm iteration and search principle. Circuits Systems and Signal Processing 39(7), 3470-3495 (2020)
33. L.J. Wang, Y. Ji, H.L. Yang, et al, Decomposition-based multiinnovation identification algorithms for a special bilinear system based on its input-output representation. International Journal of Robust and Nonlinear Control 30(9), 3607-3623 (2020)
34. J. Chen, Q.Y. Shen, J.X. Ma, Y.J. Liu, Stochastic average gradient algorithm for multirate FIR models with varying time delays using self-organizing maps. International Journal of Adaptive Control and Signal Processing 34(7), 955-970 (2020)
35. J. Pan, H. Ma, X. Zhang, et al, Recursive coupled projection algorithms for multivariable output-error-like systems with coloured noises. IET Signal Processing 14(7), 455-466 (2020)
36. Y. Ji, C. Zhang, Z. Kang, et al, Parameter estimation for block-oriented nonlinear systems using the key term separation. International Journal of Robust and Nonlinear Control 30(9), 3727-3752 (2020)
37. J. Chen, Y. Zhang, Q.M. Zhu, et al, Artken based modified Kalman filtering stochastic gradient algorithm for dual-rate nonlinear models. Journal of the Franklin Institute 356(8), 4732-4746 (2019)
38. L. Xu, F. Ding, Recursive least squares and multi-innovation stochastic gradient parameter estimation methods for signal modeling. Circuits, Systems and Signal Processing 36(4), 1735-1753 (2017)
39. M.H. Li, X.M. Liu, F. Ding, The maximum likelihood least squares based iterative estimation algorithm for bilinear systems with autoregressive moving average noise. Journal of the Franklin Institute 354(12), 4861-4881 (2017)
40. P.C. Young, Refined instrumental variable estimation: Maximum likelihood optimization of a unified Box-CoxJenkins model. Automatica 52, 35-46 (2015)