Nonlinear Dynamics manuscript No.
(will be inserted by the editor)

Iterative state and parameter estimation algorithms for bilinear state-space systems by using the block matrix inversion and the hierarchical principle

Siyu Liu · Feng Ding · Erfu Yang

Submitted: March 10, 2021

Abstract This paper is concerned with the identification of the bilinear systems in the state-space form. The parameters to be identified of the considered systems are coupled with the unknown states, which makes the identification problem difficult. To deal with such a difficulty, the iterative estimation theory is considered to derive the joint parameter and state estimation algorithm. Specifically, a moving data window least squares-based iterative (MDW-LSI) algorithm is derived to estimate the parameters of the systems by using the window data. and the unknown states are estimated by a bilinear state estimator. Furthermore, in order to improve the computational efficiency, a matrix decomposition-based MDW-LSI algorithm and a hierarchical MDW-LSI algorithm are developed according to the block matrix and the hierarchical identification principle. Finally, the computational efficiency is discussed and the numerical examples are employed to test the proposed approaches.

Keywords Nonlinear system · Bilinear system · Moving data window · Block matrix inversion · Hierarchical identification · Parameter estimation

1 Introduction

Nonlinear systems widely exist in practical industry areas. There has been great interest in the modeling, analysis, synthesis, and control of nonlinear systems which includes various descriptive models, such as neural networks [1], nonlinear time series models [2], rational models [3], and Hammerstein-Wiener structures, to name just some examples. As a particular class of nonlinear systems [4, 5], bilinear systems have relatively simple model structures but can approximately describe some complex dynamic processes such as nuclear fission, heat transfer and automobile braking systems. Specifically, the output of the chemical process was affected by the product of the external input and the system state, where the external input was the flow rate, and the system state was the temperature. Motivated by the practicality and importance of bilinear systems, the studies of bilinear systems become wider in the parameter identification and state estimation [6].

Bilinear modelling approaches such as Volterra series, Hartley-based modulating functions and Walsh functions have been widely studied. Estimating the parameters of bilinear systems is

S.Y. Liu · F. Ding (Corresponding author)
Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, People's Republic of China
E-mail: fding@jiangnan.edu.cn
ORCID: https://orcid.org/0000-0002-2721-2025

S.Y. Liu
E-mail: syliu12@126.com

E.F. Yang
Space Mechatronic Systems Technology Laboratory, University of Strathclyde, Glasgow G1 1XJ, Scotland, United Kingdom
E-mail: erfu.yang@strath.ac.uk

more challenging then estimating the parameters of linear systems for the reason that there are coupling terms of unknown states and parameters. For purpose of solving this problem, deriving the input-output representation of bilinear systems is a good way [7–9]. This approach eliminates the need to consider the effect of unknown states on the parameter estimation by removing the state variables. In addition, the subspace identification algorithm for the multivariable bilinear system was extended [10], but the bilinear nature results in the complicated calculations and heavy computational cost. For this issue, Verdult and Verhaegen presented a kernel method that depended on the dimensions of the kernel matrix [11]. The objective of this paper is to propose the estimation algorithms that can reduce the computational cost.

For the systems with large dimensions, the number of variables and the number of the parameters to be estimated increase, which makes the calculation of the estimation method increase sharply, so that the implementation of the algorithm is greatly difficult. The matrix decomposition is an effective way to reduce the computational cost by using the block matrix inversion, which is an optimization of the calculation process without changing the identification model. In addition to the matrix decomposition, the hierarchical identification principle is also applied in the field of system identification to deal with the problem that the algorithm is difficult to be performed due to the large amount of computations. Based on the combination of the hierarchical identification principle and various estimation strategies, some new identification algorithms are proposed to identify different systems [12–14], which can be applied to multivariable systems [15], nonlinear systems [16] and dual-rate stochastic systems [17]. The basic idea is to transform the original identification model into several sub-models with small sizes and to identify these sub-models by an interactive way. The two-stage iterative algorithms dealt with the variable-gain nonlinearity problem by decomposing the original model into two parts [18, 19].

When a batch of sampled data is collected, the iterative algorithm can use all the data to estimate the parameters in one iteration calculation [20]. Hence, increasing the data utilization by multiple iterations, the iterative algorithms can usually be applied in cases where the parameter estimation accuracy is required to be higher [21–23]. On the basis of the least squares-based iterative (LSI) algorithm in the previous work [24], this paper improves the algorithm performance by combining the LSI algorithm and moving data window technique. The moving data window means that the data are updated over time by adding the new observations and eliminating the old ones. This proposed algorithm uses both the past data and the current data at each step. Different from the gradient-based iterative algorithm in [25], the algorithm proposed in this paper can obtain high-precision parameter estimates based on the least squares principle compared with the gradient search that only needs to solve the first derivative. Thus, this paper presents a least squares-based iterative algorithm over moving data window (the MDW-LSI algorithm for short), which can improve the estimation accuracy. Moreover, in order to reduce the computational cost when calculating the inverse of the covariance matrix for the MDW-LSI algorithm, two computationally efficient algorithms are presented from different aspects by introducing the block matrix approach and the hierarchical identification principle. The main contributions are summarized as follows.

- Exploiting the least squares principle and the moving data window technique, we present the MDW-LSI algorithm to estimate the parameters. As for the unmeasurable states, a state estimator is designed. Combining the MDW-LSI algorithm and the new state estimator, a state estimator-based MDW-LSI algorithm is obtained to realize the joint state and parameter estimation.

- Using the hierarchical identification principle, we decompose the identification model into two sub-models and derive a hierarchical MDW-LSI (H-MDW-LSI) algorithm for identifying the two sub-models with the associate items and improving the computational efficiency.

- Introducing the inverse of the block matrix, we transform the matrix in the MDW-LSI algorithm and derive the matrix decomposition-based MDW-LSI (MD-MDW-LSI) algorithm for the purpose of reducing the computational complexity.

- The computational efficiencies of the proposed state estimator-based MD-MDW-LSI algorithm and the state estimator-based H-MDW-LSI algorithm are shown compared with that of the state estimator-based MDW-LSI algorithm.

The outlines of this paper are organized as follows. Section 2 derives the identification model of bilinear systems. Section 3 proposes the state estimator-based MDW-LSI algorithm according to the iterative estimation theory. In Sections 4 and 5, an MD-MDW-LSI algorithm and an H-MDW-LSI algorithm are derived to improve the computational efficiency. The computational complexity of these algorithms are discussed in Section 6. The numerical examples in Section 7 demonstrate the effectiveness of the proposed algorithms. Finally, some concluding remarks are given in Section 8.

2 System description and identification model

Some notations used throughout this paper are first introduced as follows.

Symbols	**Meaning**
$\hat{\boldsymbol{\theta}}_{t,k}$	The estimate of the parameter vector $\boldsymbol{\theta}$ at iteration k and time t.
T	The transpose of the vector or matrix.
$\mathbf{1}_n$	An n-dimensional column vector whose entries are all 1.
z	A unit forward shift operator like $z\boldsymbol{x}_t=\boldsymbol{x}_{t+1}$ and $z^{-1}\boldsymbol{x}_t=\boldsymbol{x}_{t-1}$.
$\boldsymbol{I}$ or $\boldsymbol{I}_n$	The identity matrix of appropriate sizes or $n\times n$.
$X:=A$	X is defined by A.
p_0	A large positive constant, e.g., $p_0=10^6$.

Consider a bilinear state-space system described by its observer canonical form as

$$\boldsymbol{x}_{t+1}=\boldsymbol{A}\boldsymbol{x}_t+\boldsymbol{F}\boldsymbol{x}_tu_t+\boldsymbol{b}u_t, \tag{1}$$

$$y_t=\boldsymbol{c}\boldsymbol{x}_t+v_t, \tag{2}$$

where $\boldsymbol{x}_t:=[x_{1,t},x_{2,t},\cdots,x_{n,t}]^{\mathrm{T}}\in\mathbb{R}^n$ is the state vector, $u_t\in\mathbb{R}$ and $y_t\in\mathbb{R}$ are the system input and output variables, v_t is an uncorrelated white noise with zero mean, i.e., $\mathrm{E}[v_t]=0$, and variance $\mathrm{E}[v_t^2]=\sigma^2$, $\mathrm{E}[v_tv_s]=0$, $t\neq s$, and $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, $\boldsymbol{F}\in\mathbb{R}^{n\times n}$, $\boldsymbol{b}\in\mathbb{R}^n$ and $\boldsymbol{c}\in\mathbb{R}^{1\times n}$ are the parameter matrices or vectors of the system,

$$\boldsymbol{A}:=\left[\begin{array}{cc}\boldsymbol{a} & \boldsymbol{I}_{n-1}\\ -a_n & \mathbf{0}\end{array}\right]\in\mathbb{R}^{n\times n},\quad \boldsymbol{F}:=\left[\begin{array}{c}\boldsymbol{f}_1\\ \boldsymbol{f}_2\\ \vdots\\ \boldsymbol{f}_n\end{array}\right]\in\mathbb{R}^{n\times n},\quad \boldsymbol{b}:=\left[\begin{array}{c}b_1\\ b_2\\ \vdots\\ b_n\end{array}\right]\in\mathbb{R}^n, \tag{3}$$

$$\boldsymbol{c}:=[1,0,\ldots,0]\in\mathbb{R}^{1\times n}, \tag{4}$$

where $\boldsymbol{a}:=[-a_1,-a_2,\cdots,-a_{n-1}]^{\mathrm{T}}\in\mathbb{R}^{n-1}$, $\boldsymbol{f}_i:=[f_{i1},f_{i2},\ldots,f_{in}]\in\mathbb{R}^{1\times n}$, $i=1,2,\cdots,n$. Assume that the dimension n of the system state is known, $y_t=0$, $u_t=0$, $\boldsymbol{x}_t=\mathbf{0}$ and $v_t=0$ for $t\leqslant 0$.

Premultiplying (1) by $\boldsymbol{z}:=[z^{-1},z^{-2},\cdots,z^{-n}]$ obtains

$$\boldsymbol{z}\boldsymbol{x}_{t+1}=\boldsymbol{z}\boldsymbol{A}\boldsymbol{x}_t+\boldsymbol{z}\boldsymbol{F}\boldsymbol{x}_tu_t+\boldsymbol{z}\boldsymbol{b}u_t. \tag{5}$$

From (5), we have the following relation:

$$x_{1,t}=\sum_{i=1}^{n}b_iu_{t-i}+\sum_{i=1}^{n}\boldsymbol{f}_i\boldsymbol{x}_{t-i}u_{t-i}-\sum_{i=1}^{n}a_ix_{1,t-i}. \tag{6}$$

From (2) and (4), we have

$$\begin{aligned}y_t&=x_{1,t}+v_t\\ &=\boldsymbol{\psi}_t^{\mathrm{T}}\boldsymbol{\theta}+v_t,\end{aligned} \tag{7}$$

where the parameter vector $\boldsymbol{\theta}$ and the information vector $\boldsymbol{\psi}_t$ are defined by

$$\boldsymbol{\theta}:=[b_1,b_2,\cdots,b_n,\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_n,a_1,a_2,\cdots,a_n]^{\mathrm{T}}\in\mathbb{R}^{n^2+2n},$$

$$\boldsymbol{\psi}_t:=[u_{t-1},u_{t-2},\cdots,u_{t-n},\boldsymbol{x}_{t-1}^{\mathrm{T}}u_{t-1},\boldsymbol{x}_{t-2}^{\mathrm{T}}u_{t-2},\cdots,\boldsymbol{x}_{t-n}^{\mathrm{T}}u_{t-n},$$

$$-x_{1,t-1},-x_{1,t-2},\cdots,-x_{1,t-n}]^{\mathrm{T}}\in\mathbb{R}^{n^2+2n}. \tag{8}$$

Equation (7) is the parameter identification model of the bilinear state-space system. This paper considers the parameter identification problem in the case that the state $\boldsymbol{x}_t$ is unknown, and presents the joint parameter and state estimation method according to the available inputs $\{u_t\}$ and outputs $\{y_t\}$, and explores the computationally efficient algorithms to reduce the computational cost.

Remark 1: It is well known that any observable and controllable state-space system can be transformed into an observable canonical form by the nonsingular linear transformation [26]. The canonical forms of the bilinear systems can reduce the number of the parameters to be identified. This paper focuses on the state and parameter estimation of the observer canonical form.

3 The state estimator-based MDW-LSI algorithm

The following derives the parameter identification algorithm using the least squares principle and the moving data window technique. It is well known that the iterative algorithm computes all sampled data at each iteration and makes efficient use of the data. The recursive algorithm can update the parameter estimates online. Differing from the methods mentioned above, this section aims to derive a state estimator-based MDW-LSI algorithm to estimate the unknown states and parameters of the considered bilinear state-space system.

Consider the newest h data (h represents the data window length) from $t-h+1$ to t, and define the stacked output vector $\boldsymbol{Y}_{h,t}$ and the stacked information matrix $\boldsymbol{\Psi}_{h,t}$ as

$$\boldsymbol{Y}_{h,t}:=\begin{bmatrix} y_t\\ y_{t-1}\\ \vdots\\ y_{t-h+1}\end{bmatrix}\in\mathbb{R}^h,\quad \boldsymbol{\Psi}_{h,t}:=\begin{bmatrix}\boldsymbol{\psi}_t^{\mathrm{T}}\\ \boldsymbol{\psi}_{t-1}^{\mathrm{T}}\\ \vdots\\ \boldsymbol{\psi}_{t-h+1}^{\mathrm{T}}\end{bmatrix}\in\mathbb{R}^{h\times(n^2+2n)}.$$

A dynamic data window criterion function is expressed as $J_1(\boldsymbol{\theta}):=\frac{1}{2}\|\boldsymbol{Y}_{h,t}-\boldsymbol{\Psi}_{h,t}\boldsymbol{\theta}\|^2$. Using the least squares principle and minimizing $J_1(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, we can obtain the least squares estimate $\hat{\boldsymbol{\theta}}_t$ of the parameter vector $\boldsymbol{\theta}$ as follows:

$$\hat{\boldsymbol{\theta}}_t=[\boldsymbol{\Psi}_{h,t}^{\mathrm{T}}\boldsymbol{\Psi}_{h,t}]^{-1}\boldsymbol{\Psi}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t}. \tag{9}$$

Faced with the difficulty that the information vector $\boldsymbol{\psi}_t$ in $\boldsymbol{\Psi}_{h,t}$ contains the unknown states $x_{1,t-i}$ and $\boldsymbol{x}_{t-i}$, $i=1,2,\ldots,n$, we cannot compute the parameter estimate $\hat{\boldsymbol{\theta}}_t$ by (9). The optimal states generally are obtained by minimizing the state estimation error covariance matrix. Motivated by this idea, a state estimator is derived for bilinear systems based on the observation information. Let $\hat{\boldsymbol{x}}_t:=[\hat{x}_{1,t},\hat{x}_{2,t},\cdots,\hat{x}_{n,t}]^{\mathrm{T}}\in\mathbb{R}^n$ be the estimate of $\boldsymbol{x}_t$. Referring to the method in [27], we construct the following bilinear state estimator:

$$\hat{\boldsymbol{x}}_{t+1}=\boldsymbol{A}\hat{\boldsymbol{x}}_t+\boldsymbol{F}\hat{\boldsymbol{x}}_tu_t+\boldsymbol{b}u_t+\boldsymbol{G}_t[y_t-\boldsymbol{c}\hat{\boldsymbol{x}}_t], \tag{10}$$

$$\boldsymbol{G}_t=[\boldsymbol{A}+\boldsymbol{F}u_t]\boldsymbol{P}_t\boldsymbol{c}^{\mathrm{T}}[1+\boldsymbol{c}\boldsymbol{P}_t\boldsymbol{c}^{\mathrm{T}}]^{-1}, \tag{11}$$

$$\boldsymbol{P}_{t+1}=[\boldsymbol{A}+\boldsymbol{F}u_t]\boldsymbol{P}_t[\boldsymbol{A}+\boldsymbol{F}u_t]^{\mathrm{T}}-\boldsymbol{G}_t\boldsymbol{c}\boldsymbol{P}_t[\boldsymbol{A}+\boldsymbol{F}u_t]^{\mathrm{T}}. \tag{12}$$

If the parameter matrices $\boldsymbol{A}$ and $\boldsymbol{F}$, and the vector $\boldsymbol{b}$ are known, we can generate the state estimate according to the state estimation algorithm in (10)–(12). Because of the unknown parameter matrices/vector $\boldsymbol{A}$, $\boldsymbol{F}$ and $\boldsymbol{b}$, the state estimate cannot be computed. To tackle this problem, the idea of the iterative estimation is used. In the following, we define the iterative index $k:=1,2,3,\cdots$, and define the estimates of $\boldsymbol{A}$, $\boldsymbol{F}$ and $\boldsymbol{b}$ at iteration k and time t as

$$\hat{\boldsymbol{A}}_{t,k}:=\begin{bmatrix}\hat{\boldsymbol{a}}_{t,k} & \boldsymbol{I}_{n-1}\\ -\hat{a}_{n,t,k} & \mathbf{0}\end{bmatrix}\in\mathbb{R}^{n\times n}, \tag{13}$$

$$\hat{\boldsymbol{a}}_{t,k}:=[-\hat{a}_{1,t,k},-\hat{a}_{2,t,k},\cdots,-\hat{a}_{n-1,t,k}]^{\mathrm{T}}\in\mathbb{R}^{n-1}, \tag{14}$$

$$\hat{\boldsymbol{F}}_{t,k}:=[\hat{\boldsymbol{f}}_{1,t,k}^{\mathrm{T}},\hat{\boldsymbol{f}}_{2,t,k}^{\mathrm{T}},\cdots,\hat{\boldsymbol{f}}_{n,t,k}^{\mathrm{T}}]^{\mathrm{T}}\in\mathbb{R}^{n\times n}, \tag{15}$$

$$\hat{\boldsymbol{b}}_{t,k} := [\hat{b}_{1,t,k}, \hat{b}_{2,t,k}, \cdots, \hat{b}_{n,t,k}]^{\mathrm{T}} \in \mathbb{R}^n. \tag{16}$$

Then, replacing $\boldsymbol{A}$, $\boldsymbol{F}$ and $\boldsymbol{b}$ in (10)–(12) with their corresponding estimates $\hat{\boldsymbol{A}}_{t,k}$, $\hat{\boldsymbol{F}}_{t,k}$ and $\hat{\boldsymbol{b}}_{t,k}$ yields the bilinear state estimation algorithm in the following,

$$\hat{\boldsymbol{x}}_{t+1,k} = \hat{\boldsymbol{A}}_{t,k}\hat{\boldsymbol{x}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}\hat{\boldsymbol{x}}_{t,k}u_t + \hat{\boldsymbol{b}}_{t,k}u_t + \boldsymbol{G}_{t,k}[y_t - \boldsymbol{c}\hat{\boldsymbol{x}}_{t,k}], \tag{17}$$

$$\boldsymbol{G}_{t,k} = [\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_t]\boldsymbol{P}_{t,k}\boldsymbol{c}^{\mathrm{T}}[1 + \boldsymbol{c}\boldsymbol{P}_{t,k}\boldsymbol{c}^{\mathrm{T}}]^{-1}, \tag{18}$$

$$\boldsymbol{P}_{t+1,k} = [\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_t]\boldsymbol{P}_{t,k}[\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_t]^{\mathrm{T}} - \boldsymbol{G}_{t,k}\boldsymbol{c}\boldsymbol{P}_{t,k}[\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_t]^{\mathrm{T}}, \tag{19}$$

where $\hat{\boldsymbol{x}}_{t,k} := [\hat{x}_{1,t,k}, \hat{x}_{2,t,k}, \cdots, \hat{x}_{n,t,k}]^{\mathrm{T}} \in \mathbb{R}^n$ is the state estimate at iteration k and time t, $\boldsymbol{P}_{t,k} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{G}_{t,k} \in \mathbb{R}^n$ are the state estimation error covariance matrix and the gain vector, respectively.

Remark 2: For setting the initial values of the above state estimation algorithm, if some specific prior knowledge of the model is known, for example, the values of the parameters are within a certain interval, the initial values can be selected within the interval. Without prior knowledge, a sufficient small vector/matrix can be used as the initial value of an algorithm. Here, the initial state vector is defined as $\hat{\boldsymbol{x}}_{1,0} = \boldsymbol{1}_n/p_0$, p_0 is a large positive constant, e.g., $p_0 = 10^6$, and the initial covariance matrix is taken as $\boldsymbol{P}_{1,0} = \boldsymbol{I}_n$, $\boldsymbol{I}_n$ is the identity matrix of appropriate sizes or $n \times n$.

According to the structure of $\boldsymbol{\psi}_t$, we define the estimate $\hat{\boldsymbol{\psi}}_{t,k}$ of the information vector $\boldsymbol{\psi}_t$ as

$$\begin{aligned}\hat{\boldsymbol{\psi}}_{t,k} := [&u_{t-1}, u_{t-2}, \cdots, u_{t-n}, \hat{\boldsymbol{x}}^{\mathrm{T}}_{t-1,k-1}u_{t-1}, \hat{\boldsymbol{x}}^{\mathrm{T}}_{t-2,k-1}u_{t-2}, \cdots, \hat{\boldsymbol{x}}^{\mathrm{T}}_{t-n,k-1}u_{t-n},\\ &-\hat{x}_{1,t-1,k-1}, -\hat{x}_{1,t-2,k-1}, \cdots, -\hat{x}_{1,t-n,k-1}]^{\mathrm{T}} \in \mathbb{R}^{n^2+2n}.\end{aligned}$$

Use the estimate $\hat{\boldsymbol{\psi}}_{t,k}$ to define the estimate $\hat{\boldsymbol{\Psi}}_{h,t,k}$ of $\boldsymbol{\Psi}_{h,t}$ as

$$\hat{\boldsymbol{\Psi}}_{h,t,k} := [\hat{\boldsymbol{\psi}}_{t,k}, \hat{\boldsymbol{\psi}}_{t-1,k}, \cdots, \hat{\boldsymbol{\psi}}_{t-h+1,k}]^{\mathrm{T}} \in \mathbb{R}^{h\times(n^2+2n)}.$$

Then, replacing the unknown stacked information matrix $\boldsymbol{\Psi}_{h,t}$ in (9) with its estimate $\hat{\boldsymbol{\Psi}}_{h,t,k}$ and combining the bilinear state estimator in (17)–(19) give the following estimation algorithm:

$$\hat{\boldsymbol{\theta}}_{t,k} = [\hat{\boldsymbol{\Psi}}^{\mathrm{T}}_{h,t,k}\hat{\boldsymbol{\Psi}}_{h,t,k}]^{-1}\hat{\boldsymbol{\Psi}}^{\mathrm{T}}_{h,t,k}\boldsymbol{Y}_{h,t}, \tag{20}$$

$$\boldsymbol{Y}_{h,t} = [y_t, y_{t-1}, \cdots, y_{t-h+1}]^{\mathrm{T}}, \tag{21}$$

$$\hat{\boldsymbol{\Psi}}_{h,t,k} = [\hat{\boldsymbol{\psi}}_{t,k}, \hat{\boldsymbol{\psi}}_{t-1,k}, \cdots, \hat{\boldsymbol{\psi}}_{t-h+1,k}]^{\mathrm{T}}, \tag{22}$$

$$\begin{aligned}\hat{\boldsymbol{\psi}}_{t,k} = [&u_{t-1}, u_{t-2}, \cdots, u_{t-n}, \hat{\boldsymbol{x}}^{\mathrm{T}}_{t-1,k-1}u_{t-1}, \hat{\boldsymbol{x}}^{\mathrm{T}}_{t-2,k-1}u_{t-2}, \cdots, \hat{\boldsymbol{x}}^{\mathrm{T}}_{t-n,k-1}u_{t-n},\\ &-\hat{x}_{1,t-1,k-1}, -\hat{x}_{1,t-2,k-1}, \cdots, -\hat{x}_{1,t-n,k-1}]^{\mathrm{T}},\end{aligned} \tag{23}$$

$$\hat{\boldsymbol{x}}_{j+1,k} = \hat{\boldsymbol{A}}_{t,k}\hat{\boldsymbol{x}}_{j,k} + \hat{\boldsymbol{F}}_{t,k}\hat{\boldsymbol{x}}_{j,k}u_j + \hat{\boldsymbol{b}}_{t,k}u_j + \boldsymbol{G}_{j,k}[y_j - \boldsymbol{c}\hat{\boldsymbol{x}}_{j,k}], \quad j = 1, 2, \cdots, t, \tag{24}$$

$$\boldsymbol{G}_{j,k} = [\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]\boldsymbol{P}_{j,k}\boldsymbol{c}^{\mathrm{T}}[1 + \boldsymbol{c}\boldsymbol{P}_{j,k}\boldsymbol{c}^{\mathrm{T}}]^{-1}, \tag{25}$$

$$\boldsymbol{P}_{j+1,k} = [\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]\boldsymbol{P}_{j,k}[\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]^{\mathrm{T}} - \boldsymbol{G}_{j,k}\boldsymbol{c}\boldsymbol{P}_{j,k}[\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]^{\mathrm{T}}, \tag{26}$$

$$\hat{\boldsymbol{A}}_{t,k} = \begin{bmatrix} \hat{\boldsymbol{a}}_{t,k} & \boldsymbol{I}_{n-1} \\ -\hat{a}_{n,t,k} & \boldsymbol{0} \end{bmatrix}, \tag{27}$$

$$\hat{\boldsymbol{a}}_{t,k} = [-\hat{a}_{1,t,k}, -\hat{a}_{2,t,k}, \cdots, -\hat{a}_{n-1,t,k}]^{\mathrm{T}}, \tag{28}$$

$$\hat{\boldsymbol{F}}_{t,k} = [\hat{\boldsymbol{f}}^{\mathrm{T}}_{1,t,k}, \hat{\boldsymbol{f}}^{\mathrm{T}}_{2,t,k}, \cdots, \hat{\boldsymbol{f}}^{\mathrm{T}}_{n,t,k}]^{\mathrm{T}}, \tag{29}$$

$$\hat{\boldsymbol{b}}_{t,k} = [\hat{b}_{1,t,k}, \hat{b}_{2,t,k}, \cdots, \hat{b}_{n,t,k}]^{\mathrm{T}}. \tag{30}$$

Equations (20)–(30) form the state estimator-based MDW-LSI (SE-MDW-LSI) algorithm. From the above derivations, it can be seen that the parameter estimate $\hat{\boldsymbol{\theta}}_{t,k}$ and the state estimate $\hat{\boldsymbol{x}}_{t,k}$ are estimated simultaneously. By adding new observation data and eliminating old observation data, the data on the moving window are updated over time, where the length of the data window is constant. The approach using both the current observations and the past observations at each iteration can improve the parameter estimation accuracy.

Remark 3: In the parameter estimation, $\hat{\boldsymbol{\theta}}_{t,0}$ can be arbitrary, we do not need to set a specific initial value. A sufficient small vector can be used as the initial value, e.g., $\hat{\boldsymbol{\theta}}_{t,0} = \boldsymbol{1}_{n^2+2n}/p_0$.

However, $\hat{\boldsymbol{x}}_{t-i,0}$, $i=1,2,\dots,n$ generally needs setting to random numbers, such initial values can guarantee that the matrix $[\hat{\boldsymbol{\Psi}}^{\mathrm{T}}_{h,t,k}\hat{\boldsymbol{\Psi}}_{h,t,k}]$ in (20) is reversible.

The following is the procedure of the state estimator-based MDW-LSI algorithm.

1. To initialize: Let $t=1$, set the moving window length h, and give a small number ε, the maximum iteration $k_{\max}$. Let $\hat{\boldsymbol{\theta}}_{t,0}=\mathbf{1}_{n^2+2n}/p_0$, $\hat{\boldsymbol{x}}_{t-i,0}=\mathbf{1}_n$, $\boldsymbol{P}_{1,0}=\boldsymbol{I}_n$, $i=1,2,\cdots,n+h$, $p_0=10^6$.
2. Let $k=1$, collect the observed data y_t and u_t, form the stacked output vector $\boldsymbol{Y}_{h,t}$ by (21).
3. Form the information vector $\hat{\boldsymbol{\psi}}_{t,k}$ by (23), and the stacked information matrix $\hat{\boldsymbol{\Psi}}_{h,t,k}$ by (22).
4. Update the parameter estimation vector $\hat{\boldsymbol{\theta}}_{t,k}$ by (20).
5. Construct the system matrices $\hat{\boldsymbol{A}}_{t,k}$ and $\hat{\boldsymbol{F}}_{t,k}$, and the system vector $\hat{\boldsymbol{b}}_{t,k}$ by (27)–(30).
6. For $j=1:t$, compute the gain vector $\boldsymbol{G}_{j,k}$ and the covariance matrix $\boldsymbol{P}_{j+1,k}$ by (25) and (26), compute the state estimate $\hat{\boldsymbol{x}}_{j+1,k}$ by (24).
7. If $k<k_{\max}$, increase k by 1 and go to Step 3; otherwise, go to the next step.
8. If $\|\hat{\boldsymbol{\theta}}_{t,k_{\max}}-\hat{\boldsymbol{\theta}}_{t,k_{\max}-1}\|>\varepsilon$, then let $\hat{\boldsymbol{\theta}}_{t+1,0}=\hat{\boldsymbol{\theta}}_{t,k_{\max}}$, go to Step 2, otherwise, obtain the parameter estimate $\hat{\boldsymbol{\theta}}_{t,k}$ and the state estimate $\hat{\boldsymbol{x}}_{t,k}$, terminate this process.

The SE-MDW-LSI algorithm in (20)–(30) estimates the parameter vector $\boldsymbol{\theta}$ and the state vector $\boldsymbol{x}_t$ in the iterative way. In order to make the algorithm more efficient, we derive an matrix decomposition based MDW-LSI algorithm using the block matrix inversion.

4 The bilinear state estimator-based MD-MDW-LSI algorithm

For the purpose of improving the computational efficiency, this section presents a matrix decomposition-based MDW-LSI (MD-MDW-LSI) algorithm for the bilinear system by applying the inverse of the block matrix. In the following, the information vector $\boldsymbol{\psi}_t$ in (8) is decomposed into two new information vectors $\boldsymbol{u}_t$ and $\boldsymbol{\chi}_t$ defined as

$$\boldsymbol{u}_t := [u_{t-1},u_{t-2},\cdots,u_{t-n}]^{\mathrm{T}}\in\mathbb{R}^n, \tag{31}$$

$$\boldsymbol{\chi}_t := [\boldsymbol{x}^{\mathrm{T}}_{t-1}u_{t-1},\boldsymbol{x}^{\mathrm{T}}_{t-2,}u_{t-2},\cdots,\boldsymbol{x}^{\mathrm{T}}_{t-n}u_{t-n},-x_{1,t-1},-x_{1,t-2},\cdots,-x_{1,t-n}]^{\mathrm{T}}\in\mathbb{R}^{n^2+n}. \tag{32}$$

Then, the information vector $\boldsymbol{\psi}_t$ and the stacked information matrix $\boldsymbol{\Psi}_{h,t}$ can be written as

$$\boldsymbol{\psi}_t := \begin{bmatrix}\boldsymbol{u}_t\\ \boldsymbol{\chi}_t\end{bmatrix}=[\boldsymbol{u}^{\mathrm{T}}_t,\boldsymbol{\chi}^{\mathrm{T}}_t]^{\mathrm{T}}\in\mathbb{R}^{n^2+2n},$$

$$\boldsymbol{\Psi}_{h,t} := \begin{bmatrix}\boldsymbol{\psi}^{\mathrm{T}}_t\\ \boldsymbol{\psi}^{\mathrm{T}}_{t-1}\\ \vdots\\ \boldsymbol{\psi}^{\mathrm{T}}_{t-h+1}\end{bmatrix}=[\boldsymbol{U}_{h,t},\boldsymbol{X}_{h,t}]\in\mathbb{R}^{h\times(n^2+2n)},$$

where the stacked information matrices $\boldsymbol{U}_{h,t}$ and $\boldsymbol{X}_{h,t}$ are defined as

$$\boldsymbol{U}_{h,t} := [\boldsymbol{u}_t,\boldsymbol{u}_{t-1},\cdots,\boldsymbol{u}_{t-h+1}]^{\mathrm{T}}\in\mathbb{R}^{h\times n}, \tag{33}$$

$$\boldsymbol{X}_{h,t} := [\boldsymbol{\chi}_t,\boldsymbol{\chi}_{t-1},\cdots,\boldsymbol{\chi}_{t-h+1}]^{\mathrm{T}}\in\mathbb{R}^{h\times(n^2+n)}. \tag{34}$$

Define the data product moment matrix

$$\begin{aligned}\boldsymbol{S}_t &:= \boldsymbol{\Psi}^{\mathrm{T}}_{h,t}\boldsymbol{\Psi}_{h,t}=\begin{bmatrix}\boldsymbol{U}^{\mathrm{T}}_{h,t}\\ \boldsymbol{X}^{\mathrm{T}}_{h,t}\end{bmatrix}[\boldsymbol{U}_{h,t},\boldsymbol{X}_{h,t}]\\ &=\begin{bmatrix}\boldsymbol{U}^{\mathrm{T}}_{h,t}\boldsymbol{U}_{h,t} & \boldsymbol{U}^{\mathrm{T}}_{h,t}\boldsymbol{X}_{h,t}\\ \boldsymbol{X}^{\mathrm{T}}_{h,t}\boldsymbol{U}_{h,t} & \boldsymbol{X}^{\mathrm{T}}_{h,t}\boldsymbol{X}_{h,t}\end{bmatrix}\in\mathbb{R}^{(n^2+2n)\times(n^2+2n)}.\end{aligned} \tag{35}$$

Then, Equation (9) can be expressed as

$$\hat{\boldsymbol{\theta}}_t=[\boldsymbol{\Psi}^{\mathrm{T}}_{h,t}\boldsymbol{\Psi}_{h,t}]^{-1}\boldsymbol{\Psi}^{\mathrm{T}}_{h,t}\boldsymbol{Y}_{h,t}=\boldsymbol{S}^{-1}_t\begin{bmatrix}\boldsymbol{U}^{\mathrm{T}}_{h,t}\boldsymbol{Y}_{h,t}\\ \boldsymbol{X}^{\mathrm{T}}_{h,t}\boldsymbol{Y}_{h,t}\end{bmatrix}. \tag{36}$$

When computing the parameter estimation vector $\hat{\boldsymbol{\theta}}_t$, it is necessary to compute the inverse of the data product moment matrix $\boldsymbol{S}_t$. This results in a larger amount of computation, and prompts us to study the new algorithm based on the matrix decomposition so as to reduce the computational cost.

Lemma 1. Suppose that $\boldsymbol{A}_1 \in \mathbb{R}^{m\times m}$ and $\boldsymbol{Q} := \boldsymbol{A}_2 - \boldsymbol{A}_{21}\boldsymbol{A}_1^{-1}\boldsymbol{A}_{12} \in \mathbb{R}^{n\times n}$ are the nonsingular matrices, then the following block matrix inversion relation holds:

$$\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_2 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}_1^{-1} + \boldsymbol{A}_1^{-1}\boldsymbol{A}_{12}\boldsymbol{Q}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_1^{-1} & -\boldsymbol{A}_1^{-1}\boldsymbol{A}_{12}\boldsymbol{Q}^{-1} \\ -\boldsymbol{Q}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_1^{-1} & \boldsymbol{Q}^{-1} \end{bmatrix}.$$

Applying Lemma 1 to Equations (35) and (36) obtains

$$\begin{aligned} \boldsymbol{S}_t^{-1} :&= \begin{bmatrix} \boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t} & \boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{X}_{h,t} \\ \boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t} & \boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{X}_{h,t} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \boldsymbol{\Omega}_t^{-1} + \boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{X}_{h,t}\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t}\boldsymbol{\Omega}_t^{-1} & -\boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{X}_{h,t}\boldsymbol{Q}_t^{-1} \\ \boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t}\boldsymbol{\Omega}_t^{-1} & \boldsymbol{Q}_t^{-1} \end{bmatrix}, \\ \boldsymbol{Q}_t &:= \boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{X}_{h,t} - \boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t}\boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{X}_{h,t} \in \mathbb{R}^{(n^2+n)\times(n^2+n)}, \qquad (37) \\ \boldsymbol{\Omega}_t &:= \boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t} \in \mathbb{R}^{n\times n}. \end{aligned}$$

Define the matrix $\boldsymbol{R}_t := \boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}} \in \mathbb{R}^{n\times h}$, and the vectors $\boldsymbol{\gamma}_t := \boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t} \in \mathbb{R}^n$ and $\boldsymbol{\beta}_t := \boldsymbol{U}_{h,t}\boldsymbol{\gamma}_t \in \mathbb{R}^h$. From (36), we have

$$\begin{aligned} \hat{\boldsymbol{\theta}}_t &= \boldsymbol{S}_t^{-1}\begin{bmatrix} \boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t} \\ \boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\gamma}_t + \boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{X}_{h,t}\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t}\boldsymbol{\gamma}_t \\ -\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t}\boldsymbol{\gamma}_t \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{X}_{h,t}\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t} \\ -\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\gamma}_t + \boldsymbol{R}_t\boldsymbol{X}_{h,t}\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{\beta}_t - \boldsymbol{R}_t\boldsymbol{X}_{h,t}\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t} \\ -\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{\beta}_t + \boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\gamma}_t + \boldsymbol{R}_t\boldsymbol{X}_{h,t}\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}(\boldsymbol{\beta}_t - \boldsymbol{Y}_{h,t}) \\ -\boldsymbol{Q}_t^{-1}\boldsymbol{X}_{h,t}^{\mathrm{T}}(\boldsymbol{\beta}_t - \boldsymbol{Y}_{h,t}) \end{bmatrix}. \qquad (38) \end{aligned}$$

From (38), it can be found that the parameter vector cannot be computed, since there are the unknown states $x_{1,t-i}$ and $\boldsymbol{x}_{t-i}$ in $\boldsymbol{X}_{h,t}$ and $\boldsymbol{Q}_t$. The identification difficulty can be solved by replacing the unknown items with their corresponding estimates at iteration $k-1$. According to the structure of $\boldsymbol{\chi}_t$, we define its estimate as

$$\begin{aligned} \hat{\boldsymbol{\chi}}_{t,k} := [&\hat{\boldsymbol{x}}_{t-1,k-1}^{\mathrm{T}}u_{t-1}, \hat{\boldsymbol{x}}_{t-2,k-1}^{\mathrm{T}}u_{t-2}, \cdots, \hat{\boldsymbol{x}}_{t-n,k-1}^{\mathrm{T}}u_{t-n}, \\ &-\hat{x}_{1,t-1,k-1}, -\hat{x}_{1,t-2,k-1}, \cdots, -\hat{x}_{1,t-n,k-1}]^{\mathrm{T}} \in \mathbb{R}^{n^2+n}. \qquad (39) \end{aligned}$$

Similarly, use the estimate $\hat{\boldsymbol{\chi}}_{t,k}$ to define the estimate of $\boldsymbol{X}_{h,t}$ as

$$\hat{\boldsymbol{X}}_{h,t,k} := [\hat{\boldsymbol{\chi}}_{t,k}, \hat{\boldsymbol{\chi}}_{t-1,k}, \cdots, \hat{\boldsymbol{\chi}}_{t-h+1,k}]^{\mathrm{T}} \in \mathbb{R}^{h\times(n^2+n)}. \qquad (40)$$

Thus, $\boldsymbol{X}_{h,t}$ in $\boldsymbol{Q}_t$ in (37) is replaced with its estimate $\boldsymbol{X}_{h,t}$, and the estimate of $\boldsymbol{Q}_t$ is obtained by

$$\hat{\boldsymbol{Q}}_{t,k} := \hat{\boldsymbol{X}}_{h,t,k}^{\mathrm{T}}\hat{\boldsymbol{X}}_{h,t,k} - \hat{\boldsymbol{X}}_{h,t,k}^{\mathrm{T}}\boldsymbol{U}_{h,t}\boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\hat{\boldsymbol{X}}_{h,t,k}.$$

Replacing $\boldsymbol{X}_{h,t}$ and $\boldsymbol{Q}_t$ in (38) with their estimates $\hat{\boldsymbol{X}}_{h,t,k}$ and $\hat{\boldsymbol{Q}}_{t,k}$ and combing the above derivations, we can summarize the MD-MDW-LSI algorithm as follows:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{t,k} &= \begin{bmatrix} \boldsymbol{\gamma}_t + \boldsymbol{R}_t\hat{\boldsymbol{X}}_{h,t,k}\hat{\boldsymbol{Q}}_{t,k}^{-1}\hat{\boldsymbol{X}}_{h,t,k}^{\mathrm{T}}(\boldsymbol{\beta}_t - \boldsymbol{Y}_{h,t}) \\ -\hat{\boldsymbol{Q}}_{t,k}^{-1}\hat{\boldsymbol{X}}_{h,t,k}^{\mathrm{T}}(\boldsymbol{\beta}_t - \boldsymbol{Y}_{h,t}) \end{bmatrix}, \qquad (41) \\ \boldsymbol{Y}_{h,t} &= [y_t, y_{t-1}, \cdots, y_{t-h+1}]^{\mathrm{T}}, \qquad (42) \\ \hat{\boldsymbol{Q}}_{t,k} &= \hat{\boldsymbol{X}}_{h,t,k}^{\mathrm{T}}\hat{\boldsymbol{X}}_{h,t,k} - \hat{\boldsymbol{X}}_{h,t,k}^{\mathrm{T}}\boldsymbol{U}_{h,t}\boldsymbol{\Omega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\hat{\boldsymbol{X}}_{h,t,k}, \qquad (43) \\ \hat{\boldsymbol{X}}_{h,t,k} &= [\hat{\boldsymbol{\chi}}_{t,k}, \hat{\boldsymbol{\chi}}_{t-1,k}, \cdots, \hat{\boldsymbol{\chi}}_{t-h+1,k}]^{\mathrm{T}}, \qquad (44) \\ \hat{\boldsymbol{\chi}}_{t,k} &= [\hat{\boldsymbol{x}}_{t-1,k-1}^{\mathrm{T}}u_{t-1}, \hat{\boldsymbol{x}}_{t-2,k-1}^{\mathrm{T}}u_{t-2}, \cdots, \hat{\boldsymbol{x}}_{t-n,k-1}^{\mathrm{T}}u_{t-n}, \end{aligned}$$

$$
\begin{aligned}
&\quad -\hat{x}_{1,t-1,k-1}, -\hat{x}_{1,t-2,k-1}, \cdots, -\hat{x}_{1,t-n,k-1}]^{\mathrm{T}}, && (45)\\
\boldsymbol{u}_t &= [u_{t-1}, u_{t-2}, \cdots, u_{t-n}]^{\mathrm{T}}, && (46)\\
\boldsymbol{U}_{h,t} &= [\boldsymbol{u}_t, \boldsymbol{u}_{t-1}, \cdots, \boldsymbol{u}_{t-h+1}]^{\mathrm{T}}, && (47)\\
\boldsymbol{\varOmega}_t &= \boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{U}_{h,t}, && (48)\\
\boldsymbol{\gamma}_t &= \boldsymbol{\varOmega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}\boldsymbol{Y}_{h,t}, && (49)\\
\boldsymbol{\beta}_t &= \boldsymbol{U}_{h,t}\boldsymbol{\gamma}_t, && (50)\\
\boldsymbol{R}_t &= \boldsymbol{\varOmega}_t^{-1}\boldsymbol{U}_{h,t}^{\mathrm{T}}, && (51)\\
\hat{\boldsymbol{A}}_{t,k} &= \begin{bmatrix} \hat{\boldsymbol{a}}_{t,k} & \boldsymbol{I}_{n-1} \\ -\hat{a}_{n,t,k} & \mathbf{0} \end{bmatrix}, && (52)\\
\hat{\boldsymbol{a}}_{t,k} &= [-\hat{a}_{1,t,k}, -\hat{a}_{2,t,k}, \cdots, -\hat{a}_{n-1,t,k}]^{\mathrm{T}}, && (53)\\
\hat{\boldsymbol{F}}_{t,k} &= [\hat{\boldsymbol{f}}_{1,t,k}^{\mathrm{T}}, \hat{\boldsymbol{f}}_{2,t,k}^{\mathrm{T}}, \cdots, \hat{\boldsymbol{f}}_{n,t,k}^{\mathrm{T}}]^{\mathrm{T}}, && (54)\\
\hat{\boldsymbol{b}}_{t,k} &= [\hat{b}_{1,t,k}, \hat{b}_{2,t,k}, \cdots, \hat{b}_{n,t,k}]^{\mathrm{T}}. && (55)
\end{aligned}
$$

Similarly, replacing $\boldsymbol{A}$, $\boldsymbol{F}$ and $\boldsymbol{b}$ with $\hat{\boldsymbol{A}}_{t,k}$, $\hat{\boldsymbol{F}}_{t,k}$ and $\hat{\boldsymbol{b}}_{t,k}$ gives the following bilinear state estimator for estimating $\boldsymbol{x}_t$:

$$
\begin{aligned}
\hat{\boldsymbol{x}}_{j+1,k} &= \hat{\boldsymbol{A}}_{t,k}\hat{\boldsymbol{x}}_{j,k} + \hat{\boldsymbol{F}}_{t,k}\hat{\boldsymbol{x}}_{j,k}u_j + \hat{\boldsymbol{b}}_{t,k}u_j + \boldsymbol{G}_{j,k}[y_j - \boldsymbol{c}\hat{\boldsymbol{x}}_{j,k}], \quad j = 1, 2, \cdots, t, && (56)\\
\boldsymbol{G}_{j,k} &= [\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]\boldsymbol{P}_{j,k}\boldsymbol{c}^{\mathrm{T}}[1 + \boldsymbol{c}\boldsymbol{P}_{j,k}\boldsymbol{c}^{\mathrm{T}}]^{-1}, && (57)\\
\boldsymbol{P}_{j+1,k} &= [\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]\boldsymbol{P}_{j,k}[\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]^{\mathrm{T}} - \boldsymbol{G}_{j,k}\boldsymbol{c}\boldsymbol{P}_{j,k}[\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]^{\mathrm{T}}. && (58)
\end{aligned}
$$

Remark 4: Equations (41)–(58) form the state estimator-based MD-MDW-LSI (SE-MD-MDW-LSI) algorithm for the joint parameter and state estimation of the bilinear system in (1) and (2). It can be found that the (1,1)th partitioned matrix $\boldsymbol{\varOmega}_t$ of $\boldsymbol{S}_t$ is invariant at each iteration (i.e., independent of k). Likewise, the matrix $\boldsymbol{R}_t$ and the vectors $\boldsymbol{\gamma}_t$ and $\boldsymbol{\beta}_t$ are independent of k, thus, they only need to be calculated at the first iteration.

To summarize, we list the steps of computing the state estimate $\hat{\boldsymbol{x}}_{t,k}$ and the parameter estimate $\hat{\boldsymbol{\theta}}_{t,k}$ by the SE-MD-MDW-LSI algorithm in (41)–(58) as follows.

1. To initialize: Let $t = 1$, set the moving window length h, and give a small number ε, the maximum iteration $k_{\max}$. Let $\boldsymbol{P}_{1,0} = \boldsymbol{I}_n$, and $\hat{\boldsymbol{\theta}}_{t,0} = \mathbf{1}_{n^2+2n}/p_0$, $\hat{\boldsymbol{x}}_{t-i,0} = \mathbf{1}_n$, $i = 1, 2, \cdots, n + h$, $p_0 = 10^6$.
2. Let $k = 1$, collect the observed data y_t and u_t, form the stacked output vector $\boldsymbol{Y}_{h,t}$ by (42), form $\boldsymbol{u}_t$ by (46) and form $\boldsymbol{U}_{h,t}$ by (47).
3. Compute $\boldsymbol{\varOmega}_t$, $\boldsymbol{\gamma}_t$, $\boldsymbol{\beta}_t$ and $\boldsymbol{R}_t$ using (48)–(51).
4. Form the information vector $\hat{\boldsymbol{\chi}}_{t,k}$ by (45), form the stacked information matrix $\hat{\boldsymbol{X}}_{h,t,k}$ by (44), and compute $\hat{\boldsymbol{Q}}_{t,k}$ by (43).
5. Update the parameter estimation vector $\hat{\boldsymbol{\theta}}_{t,k}$ by (41).
6. Construct the system matrices $\hat{\boldsymbol{A}}_{t,k}$ and $\hat{\boldsymbol{F}}_{t,k}$, and the vector $\hat{\boldsymbol{b}}_{t,k}$ using (52)–(55).
7. For $j = 1 : t$, compute the gain vector $\boldsymbol{G}_{j,k}$ by (57) and the covariance matrix $\boldsymbol{P}_{j+1,k}$ by (58), compute the state estimate $\hat{\boldsymbol{x}}_{j+1,k}$ by (56).
8. If $k < k_{\max}$, increase k by 1 and go to Step 4; otherwise, go to the next step.
9. If $\|\hat{\boldsymbol{\theta}}_{t,k_{\max}} - \hat{\boldsymbol{\theta}}_{t,k_{\max}-1}\| > \varepsilon$, then let $\hat{\boldsymbol{\theta}}_{t+1,0} = \hat{\boldsymbol{\theta}}_{t,k_{\max}}$, go to Step 2, otherwise, obtain the parameter estimate $\hat{\boldsymbol{\theta}}_{t,k}$ and the state estimate $\hat{\boldsymbol{x}}_{t,k}$, terminate this process.

Remark 5: When the dimension of the involved covariance matrix is large, the computational cost is large for obtaining its inverse matrix. For the MD-MDW-LSI algorithm, the information matrix is decomposed into two parts, one part is placed outside the iterative loop, whose inverse matrix only needs to be calculated once. The dimension of the data product moment matrix involved in the other part becomes small. Thus, the MD-MDW-LSI algorithm can improve the computational efficiency by solving the inversion of the covariance matrix.

5 The bilinear state estimator-based H-MDW-LSI algorithm

In the previous section, we use the matrix decomposition to improve the computational efficiency. Here, we apply the model decomposition to reduce the computational cost based on the hierarchical principle. Specifically, the identification model in (7) is decomposed into two sub-models with smaller dimensions and fewer variables, and the sub-models contain the associated items with each other. A hierarchical MDW-LSI (H-MDW-LSI) algorithm is derived. The details are as follows.

Define the parameter vectors and information vectors as

$$\begin{aligned}\boldsymbol{\beta} &:= [b_1, b_2, \cdots, b_n]^{\mathrm{T}} \in \mathbb{R}^n,\\ \boldsymbol{\gamma} &:= [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_n, a_1, a_2, \cdots, a_n]^{\mathrm{T}} \in \mathbb{R}^{n^2+n},\\ \boldsymbol{u}_t &:= [u_{t-1}, u_{t-2}, \cdots, u_{t-n}]^{\mathrm{T}} \in \mathbb{R}^n,\\ \boldsymbol{\chi}_t &:= [\boldsymbol{x}_{t-1}^{\mathrm{T}} u_{t-1}, \boldsymbol{x}_{t-2,}^{\mathrm{T}} u_{t-2}, \cdots, \boldsymbol{x}_{t-n}^{\mathrm{T}} u_{t-n}, -x_{1,t-1}, -x_{1,t-2}, \cdots, -x_{1,t-n}]^{\mathrm{T}} \in \mathbb{R}^{n^2+n}.\end{aligned}$$

Define two intermediate variables:

$$\begin{aligned}y_{u,t} &:= y_t - \boldsymbol{\chi}_t^{\mathrm{T}} \boldsymbol{\gamma},\\ y_{x,t} &:= y_t - \boldsymbol{u}_t^{\mathrm{T}} \boldsymbol{\beta}.\end{aligned}$$

According to the hierarchical identification principle, Equation (7) can be divided into the following two sub-models:

$$y_{u,t} := \boldsymbol{u}_t^{\mathrm{T}} \boldsymbol{\beta} + v_t, \tag{59}$$

$$y_{x,t} := \boldsymbol{\chi}_t^{\mathrm{T}} \boldsymbol{\gamma} + v_t. \tag{60}$$

For the sub-models in (59) and (60), one contains the parameter vector $\boldsymbol{\beta}$ and the information vector $\boldsymbol{u}_t$, and the other contains $\boldsymbol{\gamma}$ and $\boldsymbol{\chi}_t$. Consider the newest h data from $t-h+1$ to t. Let $\hat{\boldsymbol{\gamma}}_{t,k} := [\hat{\boldsymbol{f}}_{1,t,k}, \hat{\boldsymbol{f}}_{2,t,k}, \cdots, \hat{\boldsymbol{f}}_{n,t,k}, \hat{a}_{1,t,k}, \hat{a}_{2,t,k}, \cdots, \hat{a}_{n,t,k}]^{\mathrm{T}} \in \mathbb{R}^{n^2+n}$ and $\hat{\boldsymbol{\beta}}_{t,k} := [\hat{b}_{1,t,k}, \hat{b}_{2,t,k}, \cdots, \hat{b}_{n,t,k}]^{\mathrm{T}} \in \mathbb{R}^n$ be the estimates of $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ at iteration k and time t. Define the following two criterion functions,

$$\begin{aligned}J_2(\boldsymbol{\beta}) &:= \frac{1}{2}\sum_{s=t-h+1}^{t} [y_{u,s} - \boldsymbol{u}_s^{\mathrm{T}} \boldsymbol{\beta}]^2,\\ J_3(\boldsymbol{\gamma}) &:= \frac{1}{2}\sum_{s=t-h+1}^{t} [y_{x,s} - \boldsymbol{\chi}_s^{\mathrm{T}} \boldsymbol{\gamma}]^2.\end{aligned}$$

According to the least squares principle, minimizing $J_2(\boldsymbol{\beta})$ and $J_3(\boldsymbol{\gamma})$ gives the least squares-based iterative relations:

$$\hat{\boldsymbol{\beta}}_{t,k} = \left[\sum_{s=t-h+1}^{t} \boldsymbol{u}_s \boldsymbol{u}_s^{\mathrm{T}}\right]^{-1} \sum_{s=t-h+1}^{t} \boldsymbol{u}_s [y_s - \boldsymbol{\chi}_s^{\mathrm{T}} \boldsymbol{\gamma} - \boldsymbol{u}_s^{\mathrm{T}} \boldsymbol{\beta}], \tag{61}$$

$$\hat{\boldsymbol{\gamma}}_{t,k} = \left[\sum_{s=t-h+1}^{t} \boldsymbol{\chi}_s \boldsymbol{\chi}_s^{\mathrm{T}}\right]^{-1} \sum_{s=t-h+1}^{t} \boldsymbol{\chi}_s [y_s - \boldsymbol{u}_s^{\mathrm{T}} \boldsymbol{\beta} - \boldsymbol{\chi}_s^{\mathrm{T}} \boldsymbol{\gamma}]. \tag{62}$$

From (61) and (62), it is obvious that the information vector $\boldsymbol{\chi}_t$ on the right-hand side contains the unknown state. Also, Equations (61) and (62) contain the unknown parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Then, we replace the states with their corresponding estimates at iteration $k-1$, and replace $\boldsymbol{\chi}_t$ with $\hat{\boldsymbol{\chi}}_{t,k}$ defined in (39). The unknown parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are replaced with $\hat{\boldsymbol{\beta}}_{t,k}$ and $\hat{\boldsymbol{\gamma}}_{t,k}$. Similar to (33) and (40), we construct the stacked input vector $\boldsymbol{U}_{h,t}$ and the stacked information matrix estimate $\hat{\boldsymbol{X}}_{h,t,k}$ using $\boldsymbol{u}_t$ and $\hat{\boldsymbol{\chi}}_t$ from $t-h+1$ to t, respectively. Then, we have

$$\hat{\boldsymbol{\beta}}_{t,k} = [\boldsymbol{U}_{h,t}^{\mathrm{T}} \boldsymbol{U}_{h,t}]^{-1} \boldsymbol{U}_{h,t}^{\mathrm{T}} [\boldsymbol{Y}_{h,t} - \boldsymbol{U}_{h,t} \hat{\boldsymbol{\beta}}_{t,k-1} - \hat{\boldsymbol{X}}_{h,t,k} \hat{\boldsymbol{\gamma}}_{t,k-1}], \tag{63}$$

$$\hat{\boldsymbol{\gamma}}_{t,k} = [\hat{\boldsymbol{X}}_{h,t,k}^{\mathrm{T}} \hat{\boldsymbol{X}}_{h,t,k}]^{-1} \hat{\boldsymbol{X}}_{h,t,k}^{\mathrm{T}} [\boldsymbol{Y}_{h,t} - \hat{\boldsymbol{X}}_{h,t,k} \hat{\boldsymbol{\gamma}}_{t,k-1} - \boldsymbol{U}_{h,t} \hat{\boldsymbol{\beta}}_{t,k-1}], \tag{64}$$

$$\boldsymbol{Y}_{h,t} = [y_t, y_{t-1}, \cdots, y_{t-h+1}]^{\mathrm{T}}, \tag{65}$$

$$\begin{aligned}
\boldsymbol{U}_{h,t} &= [\boldsymbol{u}_t, \boldsymbol{u}_{t-1}, \cdots, \boldsymbol{u}_{t-h+1}]^{\mathrm{T}}, && (66)\\
\hat{\boldsymbol{X}}_{h,t,k} &= [\hat{\boldsymbol{\chi}}_{t,k}, \hat{\boldsymbol{\chi}}_{t-1,k}, \cdots, \hat{\boldsymbol{\chi}}_{t-h+1,k}]^{\mathrm{T}}, && (67)\\
\hat{\boldsymbol{\chi}}_{t,k} &= [\hat{\boldsymbol{x}}^{\mathrm{T}}_{t-1,k-1}u_{t-1}, \hat{\boldsymbol{x}}^{\mathrm{T}}_{t-2,k-1}u_{t-2}, \cdots, \hat{\boldsymbol{x}}^{\mathrm{T}}_{t-n,k-1}u_{t-n},\\
&\quad -\hat{x}_{1,t-1,k-1}, -\hat{x}_{1,t-2,k-1}, \cdots, -\hat{x}_{1,t-n,k-1}]^{\mathrm{T}}, && (68)\\
\boldsymbol{u}_t &= [u_{t-1}, u_{t-2}, \cdots, u_{t-n}]^{\mathrm{T}}, && (69)\\
\hat{\boldsymbol{A}}_{t,k} &= \begin{bmatrix} -\hat{a}_{1,t,k} & 1 & 0 & \cdots & 0\\ -\hat{a}_{2,t,k} & 0 & 1 & \ddots & \vdots\\ \vdots & \vdots & \ddots & \ddots & 0\\ -\hat{a}_{n-1,t,k} & 0 & \cdots & 0 & 1\\ -\hat{a}_{n,t,k} & 0 & \cdots & 0 & 0 \end{bmatrix}, && (70)\\
\hat{\boldsymbol{F}}_{t,k} &= [\hat{\boldsymbol{f}}^{\mathrm{T}}_{1,t,k}, \hat{\boldsymbol{f}}^{\mathrm{T}}_{2,t,k}, \cdots, \hat{\boldsymbol{f}}^{\mathrm{T}}_{n,t,k}]^{\mathrm{T}}, && (71)\\
\hat{\boldsymbol{b}}_{t,k} &= [\hat{b}_{1,t,k}, \hat{b}_{2,t,k}, \cdots, \hat{b}_{n,t,k}]^{\mathrm{T}}, && (72)\\
\hat{\boldsymbol{x}}_{j+1,k} &= \hat{\boldsymbol{A}}_{t,k}\hat{\boldsymbol{x}}_{j,k} + \hat{\boldsymbol{F}}_{t,k}\hat{\boldsymbol{x}}_{j,k}u_j + \hat{\boldsymbol{b}}_{t,k}u_j + \boldsymbol{G}_{j,k}[y_j - \boldsymbol{c}\hat{\boldsymbol{x}}_{j,k}], \quad j = 1, 2, \cdots, t, && (73)\\
\boldsymbol{G}_{j,k} &= [\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]\boldsymbol{P}_{j,k}\boldsymbol{c}^{\mathrm{T}}[1 + \boldsymbol{c}\boldsymbol{P}_{j,k}\boldsymbol{c}^{\mathrm{T}}]^{-1}, && (74)\\
\boldsymbol{P}_{j+1,k} &= [\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]\boldsymbol{P}_{j,k}[\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]^{\mathrm{T}} - \boldsymbol{G}_{j,k}\boldsymbol{c}\boldsymbol{P}_{j,k}[\hat{\boldsymbol{A}}_{t,k} + \hat{\boldsymbol{F}}_{t,k}u_j]^{\mathrm{T}}. && (75)
\end{aligned}$$

Equations (63)–(75) form the state estimator-based H-MDW-LSI (SE-H-MDW-LSI) algorithm for identifying the bilinear system. The methods proposed in this paper can combine other identification approaches [28–31] to study the parameter estimation problems of other linear systems and nonlinear systems [32–35] and can be applied to other engineering systems.

Remark 6: For the H-MDW-LSI algorithm, the identification model is decomposed into two sub-models, and the dimensions of the covariance matrices in sub-models are reduced, which can improve the computational efficiency compared with the MDW-LSI algorithm. Theoretically, the computational cost will decrease as the number of sub-models increases.

Remark 7: For setting the initial values of the SE-H-MDW-LSI algorithm, if we do not have any prior knowledge, a sufficient small vector/matrix can be used as the initial value of an algorithm. Here, the initial state vector is defined as $\hat{\boldsymbol{x}}_{1,0} = \mathbf{1}_n/p_0$, p_0 is a large positive constant, e.g., $p_0 = 10^6$, and the initial covariance matrix is taken as $\boldsymbol{P}_{1,0} = \boldsymbol{I}_n$. In the parameter estimation algorithm, we take $\hat{\boldsymbol{\beta}}_{t,0} = \mathbf{1}_n/p_0$, $\hat{\gamma}_{t,0} = \mathbf{1}_{n^2+n}/p_0$. However, $\hat{\boldsymbol{x}}_{t-i,0}$ in the information vector $\hat{\boldsymbol{\chi}}_{t,k}$ in $\hat{\boldsymbol{X}}_{h,t,k}$ generally needs setting to random numbers, such initial values can guarantee that the matrix $[\hat{\boldsymbol{X}}^{\mathrm{T}}_{h,t,k}\hat{\boldsymbol{X}}_{h,t,k}]$ in (64) is reversible.

6 The computational efficiency

Computational efficiency is an important consideration for the identification algorithms. Using the concept of the floating point is a good choice to measure the computational cost of an identification algorithm. One floating point operation can be thought of as an flop, the flop operations are represented by the sum of the number of additions and the number of multiplications [36]. To simplify the expression, a subtraction is considered an add, and a division is considered a multiplication. Here, for three different iterative algorithms, by comparing the flop operations at each iteration, the algorithm with high computational efficiency can be measured.

The following discusses the computational cost involved in the MDW-LSI algorithm, the MD-MDW-LSI algorithm and the H-MDW-LSI algorithm, and makes a comparison in detail. The multiplications and adds of the MDW-LSI algorithm at each iteration are shown in Table 1. The computational cost of the constant matrix of the MD-MDW-LSI algorithm is shown in Table 2, and the total computational cost is shown in Table 3, Table 4 shows the computational cost of the H-MDW-LSI algorithm ($n_0 := n^2 + 2n$, $n_1 = n_0 - n = n^2 + n$, n is the dimension of the system state vector, h stands for the moving data length).

Table 1 The computational cost of the MDW-LSI algorithm

Expressions	Number of multiplications	Number of adds
$\mathcal{H}_k = \hat{\boldsymbol{\Psi}}^{\mathrm{T}}_{h,t,k}\hat{\boldsymbol{\Psi}}_{h,t,k} \in \mathbb{R}^{n_0\times n_0}$	n_0^2h	$n_0^2(h-1)$
$\mathcal{R}_k = \hat{\boldsymbol{\Psi}}_{h,t,k}\boldsymbol{Y}_{h,t} \in \mathbb{R}^{n_0}$	n_0h	$n_0(h-1)$
$\hat{\boldsymbol{\theta}}_k = \mathcal{H}'_k\mathcal{R}_k \in \mathbb{R}^{n_0}$	n_0^2	$n_0(n_0-1)$
$\mathcal{H}'_k = \mathcal{H}_k^{-1} \in \mathbb{R}^{n_0\times n_0}$	n_0^3	$n_0^3-n_0^2$
Sum	$n_0^3+n_0^2+(n_0^2+n_0)h$	$n_0^3-n_0^2-2n_0+(n_0^2+n_0)h$
Total flop	$N_1 := 2n_0^3-2n_0+(2n_0^2+2n_0)h$	

Table 2 The computational cost of the constant matrices

Expressions	Number of multiplications	Number of adds
$\boldsymbol{\Omega}_t = \boldsymbol{U}^{\mathrm{T}}_{h,t}\boldsymbol{U}_{h,t} \in \mathbb{R}^{n\times n}$	n^2h	$n^2(h-1)$
$\boldsymbol{\Omega}'_t = \boldsymbol{\Omega}_t^{-1} \in \mathbb{R}^{n\times n}$	n^3	n^3-n^2
$\boldsymbol{R}_t = \boldsymbol{U}'_t\boldsymbol{U}^{\mathrm{T}}_{h,t} \in \mathbb{R}^{n\times h}$	n^2h	$n(n-1)h$
$\boldsymbol{\gamma}_t = \boldsymbol{R}_t\boldsymbol{Y}_{h,t} \in \mathbb{R}^{n}$	nh	$n(h-1)$
$\boldsymbol{\beta}_t = \boldsymbol{U}_{h,t}\boldsymbol{\gamma}_t \in \mathbb{R}^{h}$	nh	$(n-1)h$
Sum	$n^3+nh^2+(3n^2+2n)h$	$n^3-n^2+(n-1)h^2+(2n^2+n)h$
Total flop	$W := 2n^3-n^2+(2n-1)h^2+(5n^2+3n)h$	

From Table 2, we notice that $\boldsymbol{\Omega}_t$, $\boldsymbol{R}_t$, $\boldsymbol{\gamma}_t$ and $\boldsymbol{\beta}_t$ are constant matrices or vectors that are independent of the iteration k, thus they only need to be calculated once (outside the iterative loop).

The flop numbers of the MDW-LSI algorithm, the MD-MDW-LSI algorithm and the H-MDW-LSI algorithm are

$$\begin{aligned}
N_1 &:= 2n_0^3-2n_0+(2n_0^2+2n_0)h,\\
N_2 &:= 2n_1^3+n_1(2n_1+4n+2)h+nn_1(2n+2n_1-1)-2n_1+h,\\
N_3 &:= 2(n_1^3+n^3)-2(n+n_1)+2(n_1^2+n^2+3n_1+3n)h.
\end{aligned}$$

The differences in the computational cost of the three algorithms are given by

$$\begin{aligned}
N_1-N_2 &= 2n_0^3-2n_0+(2n_0^2+2n_0)h-2n_1^3-nn_1(2n+2n_1-1)\\
&\quad +2n_1-n_1(2n_1+4n+2)h-h\\
&= 4n^5+12n^4+11n^3+n^2-2n+(2n^2+2n-1)h>0,\\
N_1-N_3 &= 2n_0^3-2n_0+(2n_0^2+2n_0)h-[2(n_1^3+n^3)-2(n+n_1)\\
&\quad +2(n_1^2+n^2+3n_1+3n)h]\\
&= 6n^5+18n^4+12n^3+(4n^3+n^2-6n)h>0.
\end{aligned}$$

Since $n \geqslant 1$ and $h \geqslant 1$, the differences N_1-N_2 and N_1-N_3 are positive. That is to say the computational cost of the MD-MDW-LSI algorithm and the H-MDW-LSI algorithm are less than that of the MDW-LSI algorithm. Taking $n=6$ and $h=20$ gives an example of the flop comparison in Table 5.

From Tables 1–5, we can draw the following conclusions:

- The MD-MDW-LSI algorithm and the H-MDW-LSI algorithm are more flop-efficient than the MDW-LSI algorithm.
- The MD-MDW-LSI algorithm saves $\frac{N_1-N_2}{N_1}=\frac{50716}{315168}\approx 16.1\%$ computational cost compared with the MDW-LSI algorithm at each iteration, and the H-MDW-LSI algorithm saves $\frac{N_1-N_3}{N_1}=\frac{89856}{315168}\approx 28.5\%$, see Table 5.
- As the dimension n of the state vector increases, the computational efficiency of the MD-MDW-LSI algorithm and the H-MDW-LSI algorithm will become more prominent than that of the MDW-LSI algorithm.

Table 3 The computational cost of the MD-MDW-LSI algorithm

Expressions	Number of multiplications	Number of adds
$\boldsymbol{\lambda}_k = \boldsymbol{R}_t\hat{\boldsymbol{X}}_{h,t,k}\boldsymbol{\gamma}_k \in \mathbb{R}^n$	$nn_1(h+1)$	$nn_1h - n$
$\boldsymbol{\gamma}_k = \hat{\boldsymbol{Q}}'_{t,k}\hat{\boldsymbol{X}}^{\mathrm{T}}_{h,t,k}(\boldsymbol{\beta}_t - \boldsymbol{Y}_{h,t}) \in \mathbb{R}^{n_1}$	$n_1^2 + n_1h$	$n_1^2 + (n_1+1)h - 2n_1$
$\hat{\boldsymbol{Q}}'_{t,k} = \hat{\boldsymbol{Q}}^{-1}_{t,k} \in \mathbb{R}^{n_1\times n_1}$	n_1^3	$n_1^3 - n_1^2$
$\hat{\boldsymbol{Q}}_{t,k} = \hat{\boldsymbol{X}}^{\mathrm{T}}_{h,t,k}\hat{\boldsymbol{X}}_{h,t,k} - \mathcal{E}_k\boldsymbol{\Omega}'_t\mathcal{E}_k^{\mathrm{T}} \in \mathbb{R}^{n_1\times n_1}$	$n_1^2h + n_1^2n + n^2n_1$	$n_1^2(h-1) + nn_1(n-1) + nn_1^2$
$\mathcal{E}_k = \hat{\boldsymbol{X}}^{\mathrm{T}}_{h,t,k}\boldsymbol{U}_{h,t} \in \mathbb{R}^{n_1\times n}$	nn_1h	$nn_1(h-1)$
$\hat{\boldsymbol{\theta}}_k = \begin{bmatrix} \boldsymbol{\gamma}_t + \boldsymbol{\lambda}_k \\ -\boldsymbol{\gamma}_k \end{bmatrix} \in \mathbb{R}^{n_0}$	0	n
Sum	$n_1^3 + n_1^2 + n_1n(1+n+n_1)$	$n_1^3 - n_1^2 - 2n_1 + nn_1(n+n_1-2)$
	$+n_1(n_1+2n+1)h$	$+n_1(n_1+2n+1)h + h$
Total flop	$N_2 := 2n_1^3 + n_1(2n_1+4n+2)h + nn_1(2n+2n_1-1) - 2n_1 + h$	

Table 4 The computational cost of the H-MDW-LSI algorithm

Expressions	Number of multiplications	Number of adds
$\mathcal{H}_{1k} := \boldsymbol{U}^{\mathrm{T}}_{h,t}\boldsymbol{U}_{h,t} \in \mathbb{R}^{n\times n}$	n^2h	$n^2(h-1)$
$\mathcal{R}_{1k} := \boldsymbol{U}^{\mathrm{T}}_{h,t}\mathcal{E}_{1k}$	nh	$n(h-1)$
$\mathcal{E}_{1k} := \boldsymbol{Y}_{h,t} - \boldsymbol{U}_{h,t}\hat{\boldsymbol{\beta}}_{t,k-1} - \hat{\boldsymbol{X}}_{h,t,k}\hat{\gamma}_{t,k-1} \in \mathbb{R}^n$	$(n+n_1)h$	$(n+n_1)h$
$\hat{\boldsymbol{\beta}}_{t,k} = \mathcal{H}'_{1k}\mathcal{R}_{1k} \in \mathbb{R}^n$	n^2	$n(n-1)$
$\mathcal{H}'_{1k} := \mathcal{H}^{-1}_{1k} \in \mathbb{R}^{n\times n}$	n^3	$n^3 - n^2$
$\mathcal{H}_{2k} := \hat{\boldsymbol{X}}^{\mathrm{T}}_{h,t,k}\hat{\boldsymbol{X}}_{h,t,k} \in \mathbb{R}^{n_1\times n_1}$	n_1^2h	$n_1^2(h-1)$
$\mathcal{R}_{2k} := \hat{\boldsymbol{X}}_{h,t,k}\mathcal{E}_{2k}$	n_1h	$n_1(h-1)$
$\mathcal{E}_{2k} := \boldsymbol{Y}_{h,t} - \hat{\boldsymbol{X}}_{h,t,k}\hat{\gamma}_{t,k-1} - \boldsymbol{U}_{h,t}\hat{\boldsymbol{\beta}}_{t,k-1} \in \mathbb{R}^{n_1}$	$(n+n_1)h$	$(n+n_1)h$
$\hat{\gamma}_{t,k} = \mathcal{H}'_{2k}\mathcal{R}_{2k} \in \mathbb{R}^{n_1}$	n_1^2	$n_1(n_1-1)$
$\mathcal{H}'_{2k} := \mathcal{H}^{-1}_{2k} \in \mathbb{R}^{n_1\times n_1}$	n_1^3	$n_1^3 - n_1^2$
Sum	$n_1^3 + n^3 + n_1^2 + n^2$	$n_1^3 + n^3 - (n_1^2 + n^2) - 2(n+n_1)$
	$+(n_1^2 + n^2 + 3n_1 + 3n)h$	$+(n_1^2 + n^2 + 3n_1 + 3n)h$
Total flop	$N_3 := 2(n_1^3 + n^3) - 2(n+n_1) + 2(n_1^2 + n^2 + 3n_1 + 3n)h$	

Table 5 The computational cost of three algorithms with $n = 6$ and $h = 20$.

Algorithms	Multiplications	Adds	Flop counts
MDW-LSI	$112896 + 2352h$	$108192 + 2352h$	315168
MD-MDW-LSI	$88200 + 2310h$	$83832 + 2311h$	264452
H-MDW-LSI	$76104 + 1896h$	$72408 + 1896h$	224352

7 Examples

In this section, three examples are given to demonstrate the effectiveness of the proposed algorithms.

Example 1: A second-order bilinear state-space system is given by

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} -0.60 & 1 \\ 0.20 & 0 \end{bmatrix} \boldsymbol{x}_t + \begin{bmatrix} 0.10 & 0.70 \\ 0.30 & 0.10 \end{bmatrix} \boldsymbol{x}_t u_t + \begin{bmatrix} 1.10 \\ 1.60 \end{bmatrix} u_t,$$

$$y_t = [1,\ 0]\boldsymbol{x}_t + v_t.$$

The parameter vector to be identified is

$$\begin{aligned}\boldsymbol{\theta} &= [a_1, a_2, m_{11}, m_{12}, m_{21}, m_{22}, b_1, b_2]^{\mathrm{T}} \\ &= [0.60, -0.20, 0.10, 0.70, 0.30, 0.10, 1.10, 1.60]^{\mathrm{T}}.\end{aligned}$$

In simulation, an independent persistent excitation signal sequence $\{u_t\}$ with zero mean and unit variance is taken as the input signal, $\{v_t\}$ is an uncorrelated white noise sequence with zero

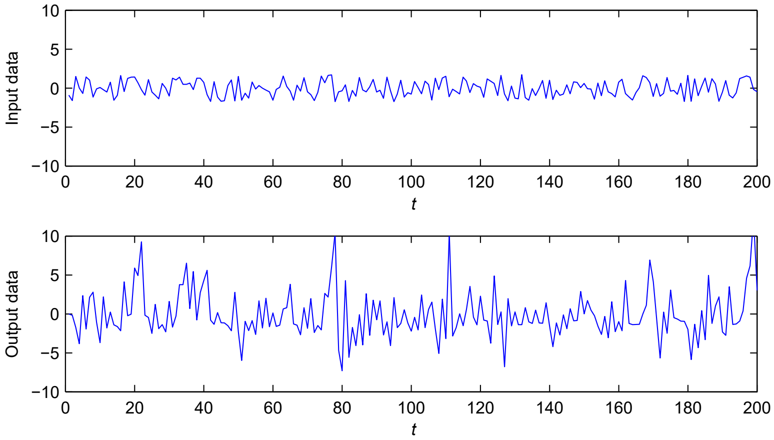

Fig. 1 The input and output signals against t

mean and variance $\sigma^2 = 0.50^2$. Take the data length $L = 500$, the moving window length $h = 100$ and let the maximum iterative number $k_{\max} = 10$. The output is generated by simulating the model with the input signal. The simulated input-output data are recorded and shown in Figure 1 (Samples 1 to 200).

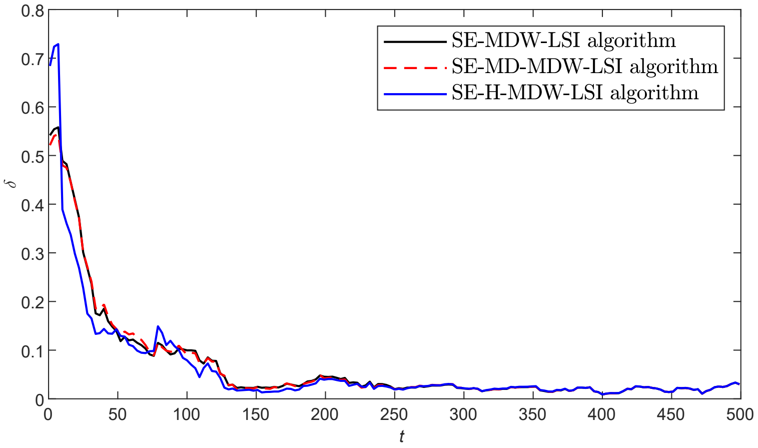

Fig. 2 The estimation errors δ against t of the three algorithms

The state estimator-based MDW-LSI algorithm in (20)–(30), the state estimator-based MD-MDW-LSI algorithm in (41)–(58) and the state estimator-based H-MDW-LSI algorithm in (63)–(75) are applied to estimate this bilinear system. The parameter estimates and errors $\delta := \|\hat{\boldsymbol{\theta}}_{t,k_{\max}} - \boldsymbol{\theta}\|/\|\boldsymbol{\theta}\| \times 100\%$ of different three algorithms are shown in Figure 2 and Table 6. From the parameter estimation error curves in Figure 2, we can see that the parameter estimation accuracies of the state estimator-based MDW-LSI algorithm and the state estimator-based MD-MDW-LSI algorithm are almost same. This means that the improved algorithm based on block matrix inversion only optimizes the calculation process and does not affect the estimation accuracy of the algorithm. Although the state estimator-based H-MDW-LSI algorithm changes the identification model, it is still effective.

Figure 3 shows the states $x_{1,t}$ and $x_{2,t}$, their estimates $\hat{x}_{1,t,k}$ and $\hat{x}_{2,t,k}$, and the state estimation errors obtained by SE-MD-MDW-LSI algorithm against t. Figure 4 shows the states $x_{1,t}$ and $x_{2,t}$, their estimates $\hat{x}_{1,t,k}$ and $\hat{x}_{2,t,k}$, and the state estimation errors obtained by SE-H-MDW-LSI algorithm against t. Since the accuracies of the SE-MDW-LSI algorithm and the SE-MD-MDW-

Table 6 The parameter estimates and errors of the three algorithms

Algorithms	t	b_1	b_2	a_1	a_2	m_{11}	m_{12}	m_{21}	m_{22}	δ (%)
SE-MDW-LSI	10	0.94816	1.76491	-0.03647	-0.12258	0.74797	0.23887	0.08345	0.11154	48.90565
	50	1.10555	1.58527	0.41894	-0.23879	0.19188	0.77313	0.15337	0.13574	12.20903
	100	0.94121	1.52677	0.58733	-0.28556	0.08649	0.77913	0.27098	0.05540	9.96132
	500	1.09561	1.54272	0.61934	-0.21338	0.09022	0.70982	0.30818	0.08700	2.99590
SE-MD-MDW-LSI	10	0.97954	1.78373	-0.01684	-0.06285	0.67929	0.21821	0.01222	0.10986	47.98185
	50	1.09604	1.57191	0.41508	-0.17832	0.18984	0.77111	0.10417	0.12786	13.56113
	100	0.95496	1.51663	0.59126	-0.24270	0.09424	0.77552	0.25399	0.06064	9.07564
	500	1.09576	1.54267	0.61938	-0.21342	0.09020	0.70984	0.30817	0.08702	2.99846
SE-H-MDW-LSI	10	1.08244	1.81832	-0.05252	-0.18048	0.46306	0.55935	-0.00163	0.02850	38.90067
	50	1.00152	1.69449	0.52533	-0.22047	0.08817	0.78055	0.11167	0.09840	11.82672
	100	0.95925	1.65800	0.55363	-0.22026	0.07830	0.69915	0.26452	0.05062	7.91768
	500	1.09599	1.54245	0.61931	-0.21353	0.09027	0.70994	0.30816	0.08683	3.00848
True values		1.10000	1.60000	0.60000	-0.20000	0.10000	0.70000	0.30000	0.10000	

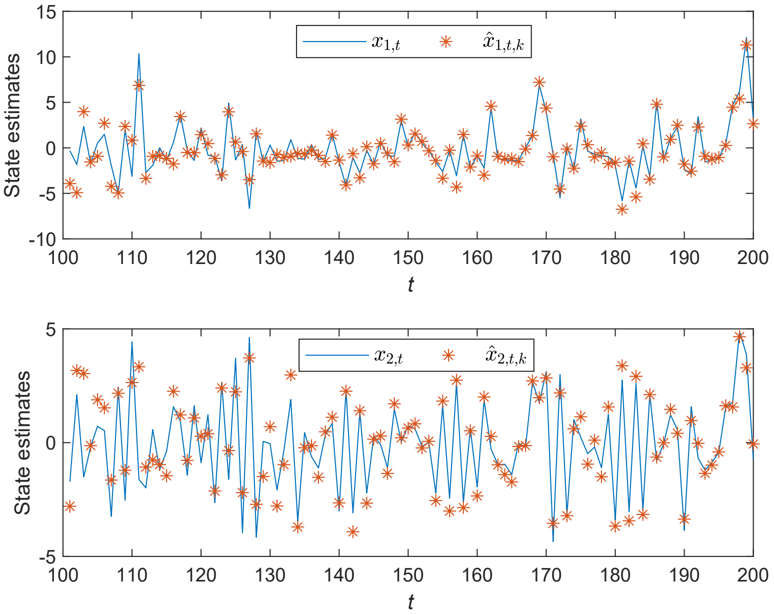

Fig. 3 The SE-MD-MDW-LSI state estimates and errors against t

LSI algorithm are almost same, only the estimation results of the SE-MD-MDW-LSI algorithm are shown. From Figures 3 and 4, it can be seen that the state estimates approach their true values with t increasing.

Taking the noise variances $\sigma^2 = 0.10^2$ and $\sigma^2 = 0.50^2$, respectively, the SE-MD-MDW-LSI and SE-H-MDW-LSI parameter estimates against t are shown in Figure 5 with $\sigma^2 = 0.10^2$ and in Figure 6 with $\sigma^2 = 0.50^2$ (solid line: the SE-MD-MDW-LSI algorithm, dots line: the SE-H-MDW-LSI algorithm), it can be found that the parameter estimates fluctuate in the beginning, but approach their true values as t increases.

To show the effectiveness of the model obtained by the MDW-LSI algorithm, we use a different data set ($L_r := 100$ samples from $t = L+1$ to $t = L+L_r$) for the model validation. The parameter estimates in the fifth row in Table 6 are used to construct the resulting model. The actual output y_t, the predicted output $\hat{y}_t$, and their errors $\hat{y}_t - y_t$ are plotted in Figure 7. It can be seen that the predicted output can track the actual output with high accuracy and small errors.

The Monte-Carlo simulations can symbolically represent the probabilistic characteristics of a method. Thus, using the Monte-Carlo simulations with m sets of different noises ($m = 1000$), the parameter estimates and the estimation deviations of the MDW-LSI algorithm are shown in Table 7, and the distributions of the parameter estimates over 1000 runs are shown in Figures 8

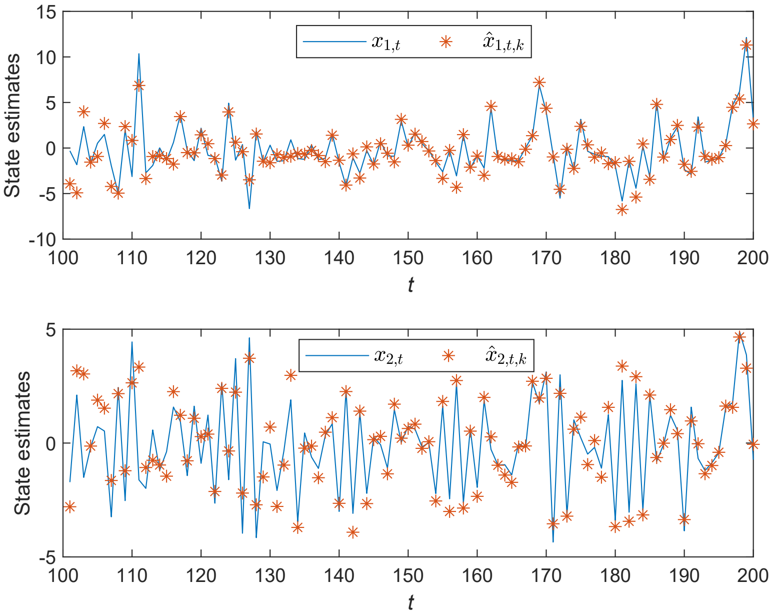

Fig. 4 The SE-H-MDW-LSI state estimates and errors against t

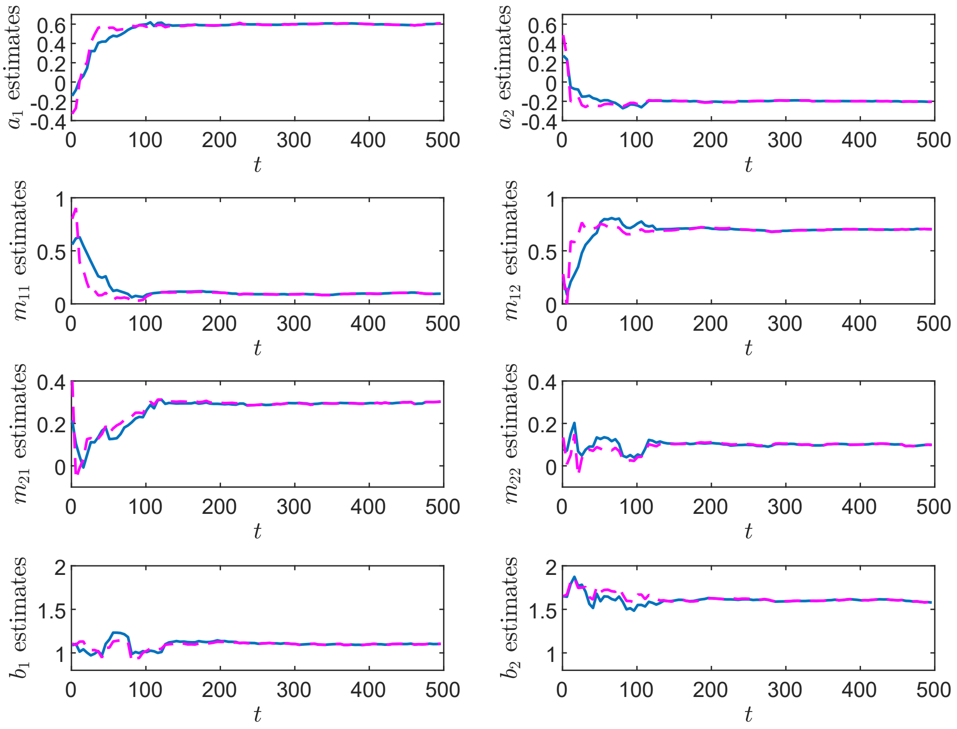

Fig. 5 The SE-MD-MDW-LSI and SE-H-MDW-LSI estimates against t ($\sigma^2 = 0.10^2$)

and 9. It can be seen that the average values of the parameter estimates are close to their true parameters.

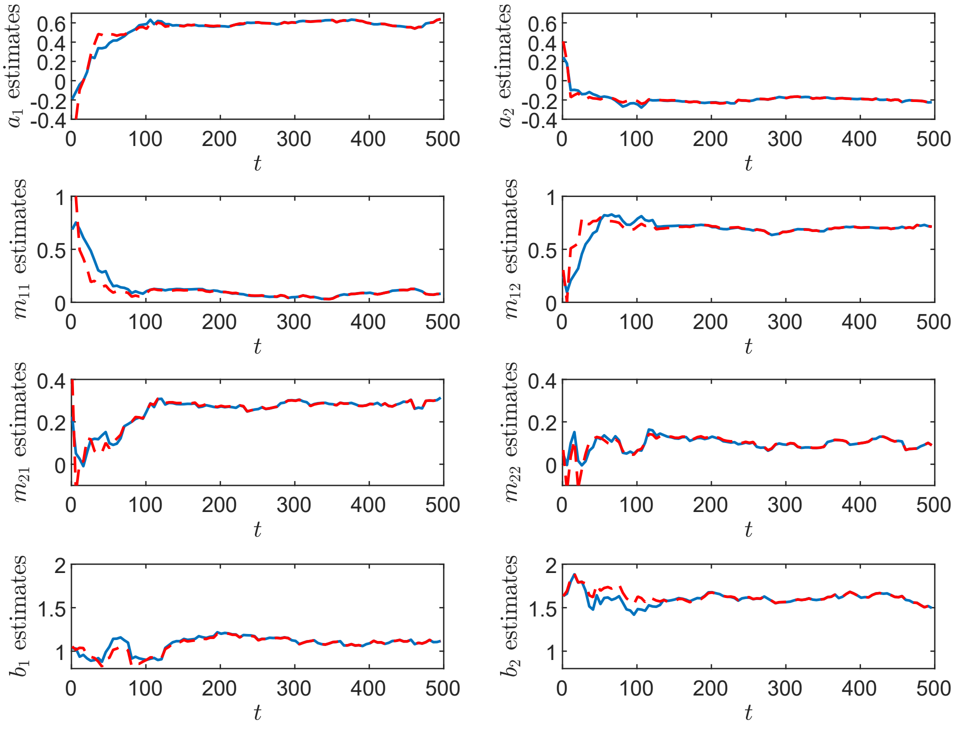

Fig. 6 The SE-MD-MDW-LSI and SE-H-MDW-LSI estimates against t ($\sigma^2 = 0.50^2$)

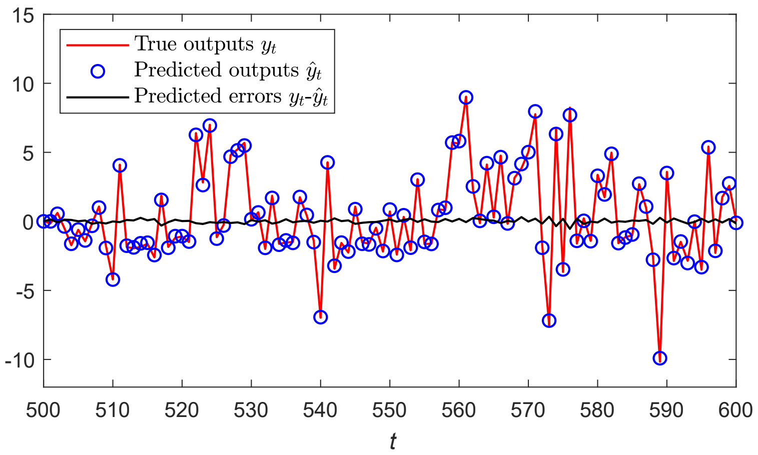

Fig. 7 The true outputs and the predicted outputs of the SE-MDW-LSI algorithm

Example 2: To test the effectiveness of the proposed algorithm for a large-scale system, a fourth-order bilinear state-space system is considered as follows:

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} -0.20 & 1 & 0 & 0 \\ 0.40 & 0 & 1 & 0 \\ 0.14 & 0 & 0 & 1 \\ -0.05 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_t + \begin{bmatrix} -0.50 & 0.30 & 0.20 & -0.10 \\ -0.03 & 0.10 & -0.08 & 0 \\ 0.40 & -0.05 & 0 & 0.30 \\ 0.05 & 0 & 0.30 & -0.20 \end{bmatrix} \boldsymbol{x}_t u_t + \begin{bmatrix} 1.10 \\ 1.60 \\ 0.80 \\ 2.20 \end{bmatrix} u_t,$$

$$y_t = [1,\ 0,\ 0,\ 0]\boldsymbol{x}_t + v_t.$$

Table 7 The SE-MDW-LSI parameter estimates and deviations for 1000 runs

k	10	50	100	500	True values
a_1	0.2367 ± 0.3532	0.593 ± 0.1076	0.5913 ± 0.0323	0.5999 ± 0.0148	0.6000
a_2	-0.1232 ± 0.2694	-0.2215 ± 0.1162	-0.2029 ± 0.0275	-0.1887 ± 0.0144	-0.2000
m_{11}	0.0746 ± 0.3271	0.0985 ± 0.0895	0.1021 ± 0.0329	0.0793 ± 0.0121	0.1000
m_{12}	0.2191 ± 0.3856	0.6569 ± 0.1283	0.6930 ± 0.0404	0.6988 ± 0.0159	0.7000
m_{21}	0.1219 ± 0.3752	0.3174 ± 0.1028	0.3003 ± 0.0319	0.2997 ± 0.0163	0.3000
m_{22}	0.0708 ± 0.4223	0.0735 ± 0.1250	0.1028 ± 0.0297	0.0997 ± 0.0140	0.1000
b_1	1.2710 ± 0.4357	1.0908 ± 0.1671	1.0976 ± 0.0621	1.0997 ± 0.0257	1.1000
b_2	1.6522 ± 0.5085	1.5888 ± 0.1767	1.6069 ± 0.0646	1.6009 ± 0.0296	1.6000

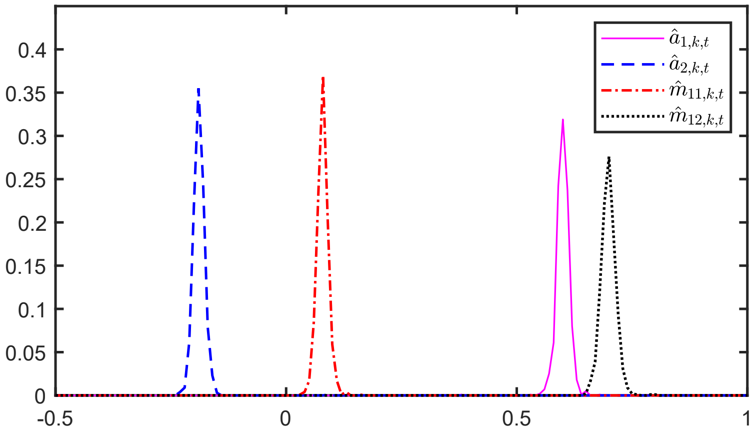

Fig. 8 The distributions of the parameter estimates over 1000 runs

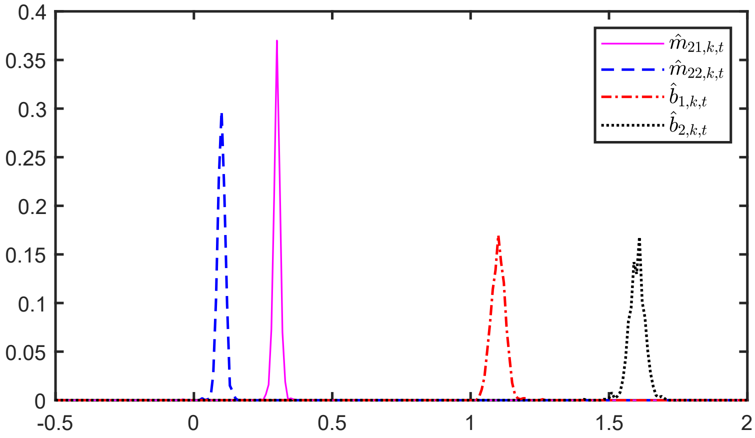

Fig. 9 The distributions of the parameter estimates over 1000 runs

In simulation, the input signal $\{u_t\}$ is the same as in Example 1. Take the data length $L = 500$, the moving window length $h = 100$ and let the maximum iterative number $k_{\max} = 10$. Under the noise variances $\sigma^2 = 0.10^2$, $\sigma^2 = 0.50^2$ and $\sigma^2 = 0.90^2$, respectively, the SE-MDW-LSI algorithm is applied to estimate the parameters and states of this fourth-order bilinear system. There are 24 parameters to be identified and their true values and estimates are shown in Table 8, and the error curves under different variances are shown in Figure 10. From Table 8 and Figure 10, it is observed that the parameter estimates are close to their true values and the estimation accuracy becomes higher with the decrease of the noise variances.

Using the parameter estimates obtained in Table 8 with the noise variance $\sigma^2 = 0.50^2$ constructs the estimated model. In order to validate the obtained model, we use another data set

Table 8 The parameter estimates and errors of the SE-MDW-LSI algorithm for the fourth-order system

σ^2	b_1=1.10	b_2=1.60	b_3=0.80	b_4=2.20	a_1=0.20	$a_2=-0.40$	$a_3=-0.14$	a_4=0.05
0.10^2	1.09536	1.58041	0.81519	2.22502	0.20692	-0.39632	-0.14516	0.04816
0.50^2	1.07582	1.49079	0.84575	2.34686	0.21831	-0.37362	-0.14987	0.04098
0.90^2	1.05209	1.40160	0.85365	2.45107	0.22133	-0.34855	-0.14268	0.03429
	$m_{11}=-0.50$	m_{12}=0.30	m_{13}=0.20	$m_{14}=-0.10$	$m_{21}=-0.03$	m_{22}=0.10	$m_{23}=-0.08$	m_{24}=0
0.10^2	-0.49775	0.30295	0.19426	-0.10655	-0.02597	0.10589	-0.08126	-0.00173
0.50^2	-0.48439	0.30202	0.17018	-0.12358	0.00613	0.11915	-0.08472	-0.00430
0.90^2	-0.46676	0.29194	0.14865	-0.13299	0.04685	0.12788	-0.08286	-0.01039
	m_{31}=0.40	$m_{32}=-0.50$	m_{33}=0	m_{34}=0.30	m_{41}=0.50	m_{42}=0	m_{43}=0.30	$m_{44}=-0.20$
0.10^2	0.40209	-0.04689	0.00297	0.29666	0.05863	-0.00255	0.30162	-0.19317
0.50^2	0.40196	-0.03134	0.01710	0.27710	0.08037	0.00079	0.29665	-0.16711
0.90^2	0.39868	-0.01345	0.02654	0.25293	0.09240	0.01390	0.28565	-0.13782

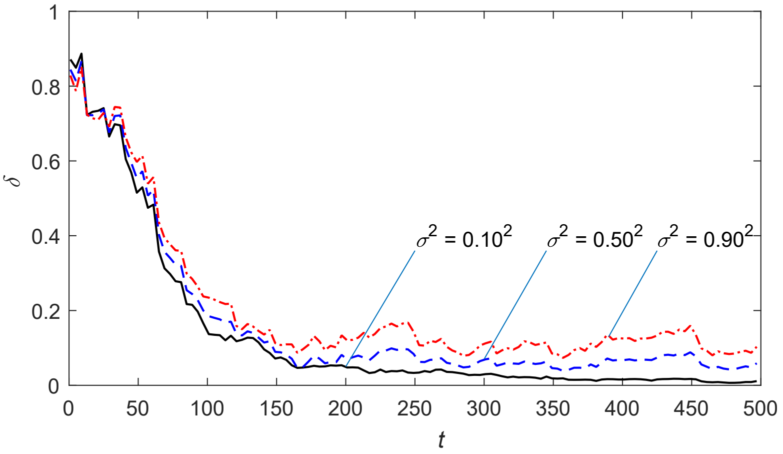

Fig. 10 The SE-MDW-LSI estimation errors δ against t with $\sigma^2 = 0.10^2$, $\sigma^2 = 0.50^2$ and $\sigma^2 = 0.90^2$

with the data length 100 for the model validation. The actual output y_t and the predicted output $\hat{y}_t$ are plotted in Figure 11. From Figure 11, we can see that the output estimates of the obtained model are close to their true outputs. In other words, the estimated model can capture the dynamics of the system. For such a fourth-order bilinear system, the identification performance of proposed algorithm is satisfactory.

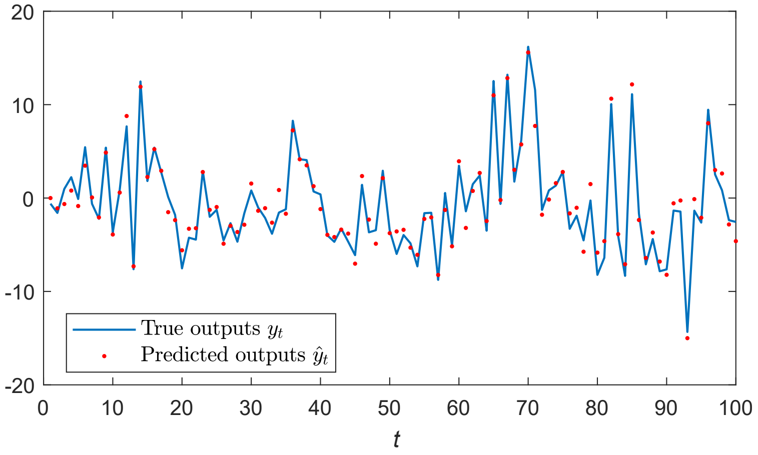

Fig. 11 The true outputs and the predicted outputs for the fourth-order system

Table 9 The steady state values of neutron kinetics model parameters

parameters	values
mean prompt neutron generation time l	10^{-3}s
the time constant of neutron delay λ	$0.1\mathrm{s}^{-1}$
delayed fraction of neutrons β	0.0075
equilibrium neutron population n_0	10^5
precursor population c_0	7.5^6

Example 3: Here, the proposed SE-MDW-LSI algorithm is tested on the neutron kinetics [37]. From a practical perspective, the purpose of the simulation is to illustrate the effectiveness of the algorithm. The neutron kinetics can be approximated by

$$\frac{d\bar{n}}{dt} = \frac{\delta_k - \beta}{l}\bar{n} + \lambda\bar{c} \tag{76}$$

$$\frac{d\bar{c}}{dt} = \frac{\beta}{l}\bar{n} - \lambda\bar{c}, \tag{77}$$

where n is the population of neutrons and c is the average population of precursor groups, δ_k is the reactivity. The values of the neutron kinetics model parameters can be seen in Table 9. At steady-state operation, the neutron population n_0 and precursor population c_0 are related by $c_0 = \frac{\beta}{l\lambda}n_0$. In order to facilitate the simulation, using the relationship of n_0 and c_0 to transform the original model, and using the forward Euler to discretize the obtained model, the model in (76) and (77) can be rewritten as

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} 1 - T_d\beta/l & T_d\beta/l \\ T_d\lambda & 1 - T_d\lambda \end{bmatrix}\boldsymbol{x}_t + \begin{bmatrix} T_d u_m/l & 0 \\ 0 & 0 \end{bmatrix}\boldsymbol{x}_t u_t, \tag{78}$$

$$y_t = [1,\ 0]\boldsymbol{x}_t + v_t, \tag{79}$$

where $x_t = [n_t, c_t]^{\mathrm{T}} = [\bar{n}_t/n_0, \bar{c}_t/c_0]^{\mathrm{T}}$ is the state variable, $u_t = \delta_k/u_m$ is the input variable, $u_m = \max\{\delta_k\}$, the output y_t is equal to n, the sample interval $T_d = 0.1s$. The model is discretized by using the forward Euler. The measurement noise variance $\sigma^2 = 0.10^2$ is added to the output equation.

According to the model parameters in Table 9, we give the input $u_t \in [-1, 1]$ and 1000 data are collected. The input and output data are used to estimate the parameters and states of this model in (78) and (79), assuming that the system model is in its observer canonical form. Since the state space model and its canonical form are equivalent, the sampled data are the same. The remaining simulation conditions are same as in Example 1. Apply the SE-MDW-LSI algorithm to identify the model and obtain a estimated model

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} 1.21766 & 1 \\ -0.24752 & 0 \end{bmatrix}\boldsymbol{x}_t + \begin{bmatrix} 0.96776 & 0.05702 \\ -0.88231 & 0.02529 \end{bmatrix}\boldsymbol{x}_t u_t. \tag{80}$$

Next, we transform the model in (78) into its observer canonical form as

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} 1.24000 & 1 \\ -0.24000 & 0 \end{bmatrix}\boldsymbol{x}_t + \begin{bmatrix} 0.90000 & 0 \\ -0.89100 & 0 \end{bmatrix}\boldsymbol{x}_t u_t. \tag{81}$$

From (80) and (81), we can see that the two models are similar, and the proposed algorithm is effective. Based on the obtained model parameter estimates in (80), another 100 sample data are collected and used for the output validation, and the results are shown in Figure 12. It can be seen from this figure that the predicted model by using the proposed algorithm can capture the dynamic system well.

8 Conclusions

This paper presents a SE-MDW-LSI algorithm for the a class of nonlinear systems to realize the joint parameter and state estimation. To increase the computational efficiency, an SE-MD-MDW-LSI algorithm and an SE-H-MDW-LSI algorithm are developed based on the block matrix

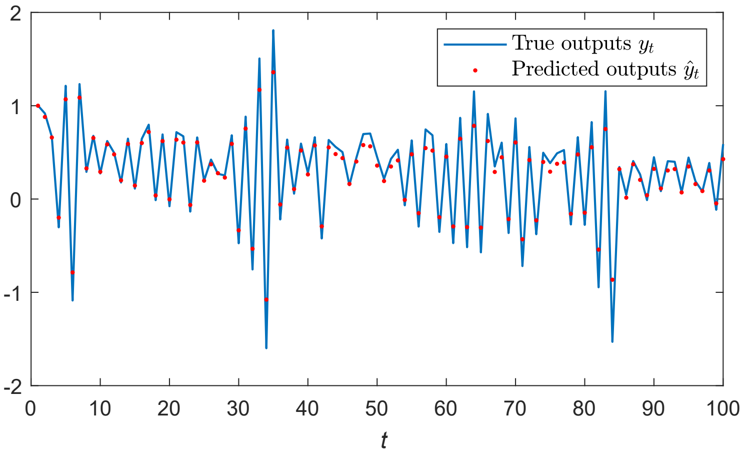

Fig. 12 The true outputs and the predicted outputs for the neutron kinetics model

inversion and the hierarchical identification principle. The simulation results demonstrate the effectiveness of the proposed algorithms, There are still several interesting topics to discuss Some system data used in modern industries may be unmeasurable or missing, and if the observation noise and the process noise are both colored noises, the problem will be more challenging and more interesting. Therefore, we will do more research in the future on the bilinear systems with missing outputs and colored noise.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 61873111), the 111 Project (B12018) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX18_1861).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Data availability statement Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks **1**(1), 4-27 (1990)
2. Xu, H., Yang, E.F.: Modeling a nonlinear process using the exponential autoregressive time series model. Nonlinear Dynamics **95**(3), 2079-2092 (2019)
3. Chen, J., Zhu, Q.M., Hu, M.F., Guo, L.X., Narayan P.: Improved gradient descent algorithms for time-delay rational state-space systems: intelligent search method and momentum method. Nonlinear Dynamics **101**(1), 361-373 (2020)
4. Ji, Y., Kang, Z., Zhang, C.: Two-stage gradient-based recursive estimation for nonlinear models by using the data filtering. International Journal of Control Automation and Systems **19**(8), 2706-2715 (2021)
5. Wang, J.W., Ji, Y., Zhang, C.: Iterative parameter and order identification for fractional-order nonlinear finite impulse response systems using the key term separation. International Journal of Adaptive Control and Signal Processing **35**(8), 1562-1577 (2021)
6. Gibson, K., Ninness, B.: Robust maximum-likelihood estimation of multivariable dynamic systems. Automatica **41**(10), 1667-1682 (2005)
7. Li, M.H., Liu, X.M.: 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)
8. Li, M.H., Liu, X.M.: 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)
9. Li, M.H., Liu, X.M.: Iterative parameter estimation methods for dual-rate sampled-data bilinear systems by means of the data filtering technique. Control Theory and Applications **15**(9), 1230-1245 (2021)
10. Wingerden, J.W., Verhaegen, M.: Subspace identification of bilinear and LPV systems for open- and closed-loop data. Automatica **45**(2), 372-381 (2009)
11. Verdult, V., Verhaegen, M.: Kernel methods for subspace identification of multivariable LPV and bilinear systems. Automatica **41**(9), 1557-1565 (2005)

12. Xu, L., Xiong, W.L., Alsaedi, A., Hayat, T.: Hierarchical parameter estimation for the frequency response based on the dynamical window data. International Journal of Control Automation and Systems **16**(4), 1756-1764 (2018)
13. Ji, Y., Zhang, C., Kang, Z., Yu, T.: Parameter estimation for block-oriented nonlinear systems using the key term separation. International Journal of Robust and Nonlinear Control **30**(9), 3727-3752 (2020)
14. Ji, Y., Kang, Z.: Three-stage forgetting factor stochastic gradient parameter estimation methods for a class of nonlinear systems. International Journal of Robust and Nonlinear Control **31**(3), 871-987 (2021)
15. Ji, Y., Jiang, X.K., Wan, L.J.: Hierarchical least squares parameter estimation algorithm for two-input Hammerstein finite impulse response systems. Journal of the Franklin Institute **357**(8), 5019-5032 (2020)
16. Wang, L.J., Ji, Y., Wan, L.J., Bu, N.: Hierarchical recursive generalized extended least squares estimation algorithms for a class of nonlinear stochastic systems with colored noise. Journal of the Franklin Institute **356**(16), 10102-10122 (2019)
17. Li, M.H., Liu, X.M.: Maximum likelihood hierarchical least squares-based iterative identification for dual-rate stochastic systems. International Journal of Adaptive Control and Signal Processing **35**(2), 240-261 (2021)
18. Fan, Y.M., Liu, X.M.: Two-stage auxiliary model gradient-based iterative algorithm for the input nonlinear controlled autoregressive system with variable-gain nonlinearity. International Journal of Robust and Nonlinear Control **30**(14), 5492-5509 (2020)
19. Liu, X.M., Fan, Y.M.: Maximum likelihood extended gradient-based estimation algorithms for the input nonlinear controlled autoregressive moving average system with variable-gain nonlinearity. International Journal of Robust and Nonlinear Control **31**(9), 4017-4036 (2021)
20. Xu, L.: 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), 141-151 (2019)
21. Xu, L.: The damping iterative parameter identification method for dynamical systems based on the sine signal measurement. Signal Processing **120**, 660-667 (2016)
22. Chen, J., Zhu, Q.M., Liu, Y.J.: Modified Kalman filtering based multi-step-length gradient iterative algorithm for ARX models with random missing outputs. Automatica **118**, 109034 (2020)
23. Ma, H., Pan, J.: Partially-coupled least squares based iterative parameter estimation for multi-variable output-error-like autoregressive moving average systems. IET Control Theory and Applications **13**(18), 3040-3051 (2019)
24. Liu, S.Y., Zhang, Y.L.: Three-stage least squares-based iterative estimation algorithms for bilinear state-space systems based on the bilinear state estimator. International Journal of Adaptive Control and Signal Processing **34**(10), 1501-1518 (2020)
25. Liu, S.Y., Zhang, Y.L., Xu, L., et al.: Extended gradient-based iterative algorithm for bilinear state-space systems with moving average noises by using the filtering technique. International Journal of Control Automation and Systems **19**(4), 1597-1606 (2021)
26. Zhang, X.: Recursive parameter identification of the dynamical models for bilinear state space systems. Nonlinear Dynamics **89**(4), 2415-2429 (2017)
27. Liu, S.Y.: Moving data window gradient-based iterative algorithm of combined parameter and state estimation for bilinear systems. International Journal of Robust and Nonlinear Control **30**(6), 2413-2429 (2020)
28. Xu, L..: Separable multi-innovation stochastic gradient estimation algorithm for the nonlinear dynamic responses of systems. International Journal of Adaptive Control and Signal Processing **34**(7), 937-954 (2020)
29. Xu, L., Chen, F.Y.: Hierarchical recursive signal modeling for multi-frequency signals based on discrete measured data. International Journal of Adaptive Control and Signal Processing **35**(5), 676-693 (2021)
30. Xu, L.: Separable recursive gradient algorithm for dynamical systems based on the impulse response signals. International Journal of Control Automation and Systems **18**(12), 3167-3177 (2020)
31. Liu, L.J., Liu, H.B.: Data filtering based maximum likelihood gradient estimation algorithms for a multivariate equation-error system with ARMA noise. Journal of the Franklin Institute **357**(9), 5640-5662 (2020)
32. Xu, L., Song, G.L.: A recursive parameter estimation algorithm for modeling signals with multi-frequencies. Circuits Systems and Signal Processing **39**(8), 4198-4224 (2020)
33. Xu, L.: Separable Newton recursive estimation method through system responses based on dynamically discrete measurements with increasing data length. International Journal of Control Automation and Systems **20**, (2022).
34. Ma, P., Wang, L.: Filtering-based recursive least squares estimation approaches for multivariate equation-error systems by using the multiinnovation theory. International Journal of Adaptive Control and Signal Processing **35**(9), 1898-1915 (2021)
35. Chen, J., Shen, Q.Y., Ma, J.X., Liu, Y.J.: 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)
36. Golub, G.H., Van Loan, C.F.: Matrix Computations (third ed.). Johns Hopkins University Press, Baltimore, MD, (1996)
37. Baheti, R.S., Mohler, R.R., Spang, H.A.: Second-order correlation method for bilinear system identification. IEEE Transactions on Automatic Control **25**(6), 1141-1146 (1980)