IET Research Journals

Regular Paper

State filtering based least squares parameter estimation for bilinear systems using the hierarchical identification principle

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Xiao Zhang[1], Feng Ding[1*], Ling Xu[1], Erfu Yang[2]

[1] 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
[2] Department of Design, Manufacture and Engineering Management, Space Mechatronic Systems Technology Laboratory, Strathclyde Space Institute, University of Strathclyde, Glasgow G1 1XJ, Scotland, United Kingdom
* E-mail: fding@jiangnan.edu.cn

Abstract: **In the revised paper, we have highlighted the changes made in Blue FONT.**
This paper presents a combined parameter and state estimation algorithm for a bilinear system described by its observer canonical state space model based on the hierarchical identification principle. The Kalman filter is known as the best state filter for linear systems, but not applicable for bilinear systems. Thus, a bilinear state observer (BSO) is designed to give the state estimates using the extremum principle. Then a BSO based recursive least squares (BSO-RLS) algorithm is developed. For comparison with the BSO-RLS algorithm, by dividing the system into three fictitious subsystems on basis of the decomposition-coordination principle, a BSO based hierarchical least squares algorithm is proposed to reduce the computation burden. Moreover, a BSO based forgetting factor recursive least squares algorithm is presented to improve the parameter tracking capability. Finally, a numerical example illustrates the effectiveness of the proposed algorithms.

1 Introduction

Parameter estimation is a significant part in system identification, and has been widely used in system analysis [1–3], system modeling [4–7], and system control [8, 9]. Since many industrial processes are complex and inherently nonlinear, nonlinear system identification has drawn much attention throughout the world [10–12]. The bilinear approach for modeling these complex processes are proven to be more precise than any other traditional linear models [13]. However, the nonlinear term existing in the bilinear model brings some challenges for bilinear system identification. During the past decades, much work has been carried out on parameter estimation for bilinear systems. For example, dos Santos et al. presented a subspace identification method for bilinear systems by treating the bilinear term as a second-order white noise process [14]. Larkowski et al. addressed the identification problem of the diagonal bilinear errors-in-variables system and extended the bias compensated least squares technique to bilinear systems [15]. Li et al. applied the polynomial transformation technique to obtain the equivalent input-output representation of the bilinear system and proposed the iterative algorithm for parameter estimation [16–18].

The recursive least squares (RLS) approach is known as the most commonly used estimation method among numerous different parameter estimation techniques. Although the RLS method offers a fast convergence rate, there exists several problems such as the increase in the computational burden and the decline in the tracking capability [19, 20]. The hierarchical identification principle is applied to decompose a bilinear system into several subsystems for parameter estimation [21]. Recently, Ma et al. proposed the modified Kalman filter hierarchical least squares algorithm for the multivariate Hammerstein system [22]. Wang et al. presented a hierarchical stochastic gradient algorithm for bilinear-in-parameter systems [23]. Chu et al. proposed a diffusion variable forgetting factor RLS algorithm on the basis of a local polynomial modeling of the time-varying systems [24]. Wang et al. presented an interval varying recursive least squares algorithm with two forgetting factors for pseudo-linear systems with missing data [25].

Nonlinear filtering techniques have attracted much attention in signal processing [26, 27] and have wide applications in many areas [28–31]. The classical Kalman filter (KF) is recognized as the best linear filter for linear systems under Gaussian noises. However, it is not suitable for nonlinear systems and bilinear systems, which promotes the development of the alternative filtering methods such as the extended KF, the unscented KF and the H-infinity filter. In the literature, Favoreel et al. considered a bilinear system as a time-varying linear system, and applied the Kalman filter to estimate the unknown states [32]. Basin et al. designed a mean-square finite-dimensional filter for the incompletely measured bilinear time-delay system over linear observations [33].

The previous work in [34] considered the state filtering and least squares estimation problem for a linear state space system. It is well known that the system identification of nonlinear systems is more difficult than that of the linear case. Moreover, for linear systems, the Kalman filtering algorithm can directly obtains the state estimates, but is not applicable for nonlinear system state filtering. Thus, this paper extends the parameter estimation and state filtering problem from linear systems to bilinear systems. In the state estimation, the bilinear state observer is designed to obtain the system states by replacing the unknown parameters with their estimates. In the parameter estimation, by replacing the unknown states with their estimates, the parameters can be identified based on the least squares principle. They form the interactive estimation. Then the bilinear state observer based recursive least squares (BSO-RLS) algorithm is proposed. Furthermore, the original system is decomposed into three subsystems by means of the decomposition-coordination principle and a bilinear state observer based hierarchical least squares (BSO-HLS) algorithm is derived to reduce computational burden. Finally, a bilinear state observer based forgetting factor recursive least squares algorithm (BSO-FF-RLS) is proposed to improve the parameter tracking capability compared with BSO-RLS algorithm.

This paper is organized as follows. Section 2 gives the identification model of the bilinear state space system and introduces the identification problems to be discussed. Section 3 derives a bilinear state observer by minimizing the state estimation error. Section 4 proposes a BSO-RLS algorithm based on the state observer. By using the hierarchical identification technique, Section 5 presents the BSO-HLS algorithm based on the extremum principle. In Section 6, the BSO-FF-RLS algorithm is proposed and the computational complexity analysis is discussed. The example is provided in Section 7 to verify the effectiveness of the proposed algorithm. Finally, concluding remarks are given in Section 8.

2 The bilinear system and its identification model

Let the expression "$A =: X$"or "$X := A$" stand for "A is defined as X". Let the superscript T denote the matrix/vector transpose, $\hat{\boldsymbol{\theta}}(t)$ denote the estimate of the parameter $\boldsymbol{\theta}$ at time t. The symbol $\boldsymbol{I}$ ($\boldsymbol{I}_n$) stands for an identity matrix of appropriate sizes ($n \times n$); z represents a unit forward shift operator: $z\boldsymbol{x}(t) = \boldsymbol{x}(t+1)$ and $z^{-1}\boldsymbol{x}(t) = \boldsymbol{x}(t-1)$. $\mathbf{1}_n$ represents an $n-$dimensional column vector whose elements are all unity.

Consider a bilinear system in Figure 1:

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{x}(t)u(t) + \boldsymbol{f}u(t), \tag{1}$$

$$y(t) = \boldsymbol{c}\boldsymbol{x}(t) + v(t), \tag{2}$$

where $\boldsymbol{x}(t) := [x_1(t), x_2(t), \ldots, x_n(t)]^{\mathrm{T}} \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}$ is the system input, $y(t) \in \mathbb{R}$ is the system output, $v(t) \in \mathbb{R}$ is an uncorrelated random noise with zero mean, and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times n}$, $\boldsymbol{f} \in \mathbb{R}^n$ and $\boldsymbol{c} \in \mathbb{R}^{1\times n}$ are the system parameter matrices/vectors. Transforming the bilinear system in (1)–(2) in its observer canonical state space model gives

$$\boldsymbol{A} := \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -a_{n-1} & 0 & \cdots & 0 & 1 \\ -a_n & 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

$$\boldsymbol{B} := \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_n \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad \boldsymbol{b}_i \in \mathbb{R}^{1\times n},$$

$$\boldsymbol{f} := [f_1, f_2, \ldots, f_n]^{\mathrm{T}} \in \mathbb{R}^n,$$

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

Referring to the method in [35, 36] and from (1)–(2), we have

$$x_1(t) = -\sum_{i=1}^n a_i x_1(t-i) + \sum_{i=1}^n \boldsymbol{b}_i \boldsymbol{x}(t-i)u(t-i) + \sum_{i=1}^n f_i u(t-i).$$

Define the information vector $\boldsymbol{\varphi}(t)$ and the system parameter vector $\boldsymbol{\theta}$ as

$$\begin{aligned}
\boldsymbol{\varphi}(t) &:= [\boldsymbol{\varphi}_x^{\mathrm{T}}(t), \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t), \boldsymbol{\varphi}_u^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{n^2+2n}, \\
\boldsymbol{\varphi}_x(t) &:= [-x_1(t-1), -x_1(t-2), \ldots, -x_1(t-n)]^{\mathrm{T}} \in \mathbb{R}^n, \\
\boldsymbol{\varphi}_{xu}(t) &:= [\boldsymbol{x}^{\mathrm{T}}(t-1)u(t-1), \boldsymbol{x}^{\mathrm{T}}(t-2)u(t-2), \ldots, \\
&\quad \boldsymbol{x}^{\mathrm{T}}(t-n)u(t-n)]^{\mathrm{T}} \in \mathbb{R}^{n^2}, \\
\boldsymbol{\varphi}_u(t) &:= [u(t-1), u(t-2), \ldots, u(t-n)]^{\mathrm{T}} \in \mathbb{R}^n, \\
\boldsymbol{\theta} &:= [\boldsymbol{a}^{\mathrm{T}}, \boldsymbol{b}^{\mathrm{T}}, \boldsymbol{f}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{n^2+2n}, \\
\boldsymbol{a} &:= [a_1, a_2, \ldots, a_n]^{\mathrm{T}} \in \mathbb{R}^n, \\
\boldsymbol{b} &:= \mathrm{col}[\boldsymbol{B}^{\mathrm{T}}] \in \mathbb{R}^{n^2}, \\
\boldsymbol{f} &:= [f_1, f_2, \ldots, f_n]^{\mathrm{T}} \in \mathbb{R}^n.
\end{aligned}$$

Then the identification model of the bilinear system in (1)–(2) can be expressed as

$$\begin{aligned}
y(t) &= \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\boldsymbol{a} + \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\boldsymbol{b} + \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\boldsymbol{f} + v(t) \\
&= \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta} + v(t).
\end{aligned} \tag{3}$$

According to the least squares principle, defining and minimizing the cost function

$$J(\boldsymbol{\theta}) := \sum_{j=1}^t [y(j) - \boldsymbol{\varphi}^{\mathrm{T}}(j)\boldsymbol{\theta}]^2$$

lead to the following recursive least squares algorithm:

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{L}(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)], \tag{4}$$

$$\boldsymbol{L}(t) = \boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)[1 + \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)]^{-1}, \tag{5}$$

$$\boldsymbol{P}(t) = [\boldsymbol{I} - \boldsymbol{L}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)]\boldsymbol{P}(t-1), \tag{6}$$

where $\boldsymbol{P}(t)$ is the covariance matrix and $\boldsymbol{L}(t) := \boldsymbol{P}(t)\boldsymbol{\varphi}(t)$ is the gain vector. Give the initial parameter as $\hat{\boldsymbol{\theta}}(0) = \mathbf{1}_{n^2+2n}/p_0$.

Remark 1. Notice that $\boldsymbol{\varphi}(t)$ involves the unknown states $\boldsymbol{x}(t)$. The recursive least squares algorithm in (4)–(6) cannot obtain the system parameters, so it is necessary to derive a combined parameter estimation and state filtering algorithm for the bilinear system. For a linear system, the Kalman filter can be applied to give its state estimates. However, the system in (1)–(2) is bilinear. Thus, we are supposed to design a bilinear state observer for state estimation and choose a suitable observer gain vector so as to minimize the state estimation error, which is similar to the requirement of the Kalman filter for the linear case. The details are in Section 3.

Remark 2. In system identification, the standard least squares method is recognized as the widely used parameter estimation method. In spite of its widespread use, it presents some potential problems in this paper and these drawbacks motivate us to study new identification methods. The bilinear system in (1)–(2) contains $n^2 + 2n$ parameters to be identified. With the dimension n increasing, the system contains more parameters and the dimension of $\boldsymbol{P}(t)$ becomes larger especially for large-scale bilinear systems. The amount of calculation for $\boldsymbol{P}(t)$ determines the entire computational load, so a high-dimensional $\boldsymbol{P}(t)$ leads to heavy computational burden. Thus, in order to reduce the calculation amount, this paper uses the hierarchical identification principle to decompose the system into several subsystems. Moreover, in order to improve the parameter tracking capability for the bilinear system, we introduce a forgetting factor in the parameter estimation algorithm and improve the algorithm activity.

IET Research Journals, pp. 1–10

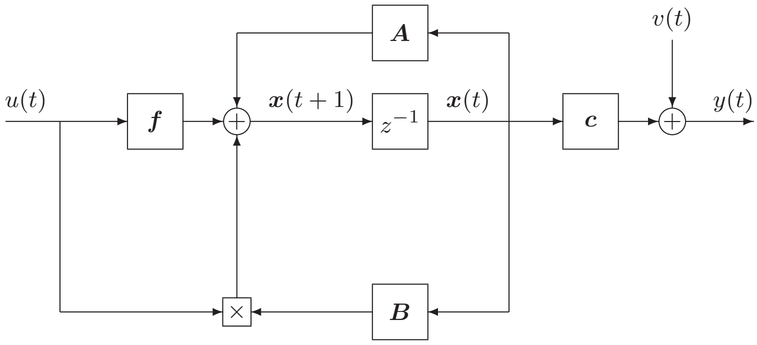

Fig. 1: The bilinear state space system

3 The bilinear state observer

It is known that the Kalman filter is the best linear state filter for the linear system, but it cannot be applied to obtain the unknown states of nonlinear systems (i.e., bilinear systems). Thus, this section presents a bilinear state observer to obtain the state estimates $\hat{\boldsymbol{x}}(t)$ of the unknown states $\boldsymbol{x}(t)$ for the bilinear system by minimizing the state estimation error.

If the parameter matrices/vector $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{f}$ are known, similar to the state observer, we construct

$$\begin{aligned}\hat{\boldsymbol{x}}(t+1) &= \boldsymbol{A}\hat{x}(t) + \boldsymbol{B}\hat{x}(t)u(t)\\ &\quad + \boldsymbol{f}u(t) + \boldsymbol{L}_x(t)[y(t) - \boldsymbol{c}\hat{\boldsymbol{x}}(t)],\end{aligned}$$

where $\boldsymbol{L}_x(t)$ is the observer gain vector. Defining the state estimation error $\tilde{\boldsymbol{x}}(t) := \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$, we have

$$\tilde{\boldsymbol{x}}(t+1) = [\boldsymbol{A} - \boldsymbol{L}_x(t)\boldsymbol{c}]\tilde{\boldsymbol{x}}(t) + \boldsymbol{B}\tilde{\boldsymbol{x}}(t)u(t) - \boldsymbol{L}_x(t)v(t). \quad (7)$$

Define the state estimation error covariance matrix

$$\boldsymbol{P}_x(t) = \mathrm{E}[\tilde{\boldsymbol{x}}(t)\tilde{\boldsymbol{x}}^{\mathrm{T}}(t)]. \quad (8)$$

The aim is to choose an optimal gain vector $\boldsymbol{L}_x(t)$ to minimize the state estimation error covariance matrix $\boldsymbol{P}_x(t+1) := \mathrm{E}[\tilde{\boldsymbol{x}}(t+1)\tilde{\boldsymbol{x}}^{\mathrm{T}}(t+1)]$. Using (7), we have

$$\begin{aligned}\boldsymbol{P}_x(t+1) &= [\boldsymbol{A} - \boldsymbol{L}_x(t)\boldsymbol{c}]\boldsymbol{P}_x(t)[\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t) + \boldsymbol{B}^{\mathrm{T}}u(t)]\\ &\quad + \boldsymbol{B}u(t)\boldsymbol{P}_x(t)[\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t) + \boldsymbol{B}^{\mathrm{T}}u(t)]\\ &\quad + \boldsymbol{L}_x(t)R_v\boldsymbol{L}_x^{\mathrm{T}}(t). \quad (9)\end{aligned}$$

It is difficult to determine $\boldsymbol{L}_x(t)$ because computing the partial derivative of $\boldsymbol{P}_x(t+1)$ in (9) with respect to $\boldsymbol{L}_x(t)$ is not easy. Here, we adopt the extremum principle to obtain the filtering gain vector $\boldsymbol{L}_x(t)$. Assume that $\boldsymbol{L}_x(t)$ is the optimal gain vector to minimize the state estimation error covariance matrix $\boldsymbol{P}_x(t+1)$, and $\boldsymbol{P}_x(t+1)$ is minimal. It is obvious that if there exists the departure $\delta\boldsymbol{L}_x(t)$ from the filtering gain vector to the optimal gain vector, the estimation error covariance matrix obtained from (9) will deviate from the minimal $\boldsymbol{P}_x(t+1)$ and reach $\boldsymbol{P}_x(t+1) + \delta\boldsymbol{P}_x(t+1)$, where $\delta\boldsymbol{P}_x(t+1)$ is the non-negative definite matrix. From (9), we find that $\boldsymbol{L}_x(t) + \delta\boldsymbol{L}_x(t)$ and $\boldsymbol{P}_x(t+1) + \delta\boldsymbol{P}_x(t+1)$ satisfy

$$\begin{aligned}&\boldsymbol{P}_x(t+1) + \delta\boldsymbol{P}_x(t+1)\\ &= \{\boldsymbol{A} - [\boldsymbol{L}_x(t) + \delta\boldsymbol{L}_x(t)]\boldsymbol{c}\}\boldsymbol{P}_x(t)\\ &\quad \times\{\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}^{\mathrm{T}}[\boldsymbol{L}_x(t) + \delta\boldsymbol{L}_x(t)]^{\mathrm{T}} + \boldsymbol{B}^{\mathrm{T}}u(t)\}\\ &\quad + \boldsymbol{B}u(t)\boldsymbol{P}_x(t)\{\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}^{\mathrm{T}}[\boldsymbol{L}_x(t) + \delta\boldsymbol{L}_x(t)]^{\mathrm{T}} + \boldsymbol{B}^{\mathrm{T}}u(t)\}\\ &\quad + [\boldsymbol{L}_x(t) + \delta\boldsymbol{L}_x(t)]R_v[\boldsymbol{L}_x(t) + \delta\boldsymbol{L}_x(t)]^{\mathrm{T}}, \quad (10)\end{aligned}$$

where $\boldsymbol{P}_x(t+1)$ and $\boldsymbol{L}_x(t)$ satisfy (9). Substituting (9) into (10) gives

$$\begin{aligned}&\delta\boldsymbol{P}_x(t+1)\\ &= -\delta\boldsymbol{L}_x(t)[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{A}^{\mathrm{T}}\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t) + \boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{B}^{\mathrm{T}}u(t)\\ &\quad - R_v\boldsymbol{L}_x^{\mathrm{T}}(t)] - [\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t) + \boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{B}^{\mathrm{T}}u(t)\\ &\quad - R_v\boldsymbol{L}_x^{\mathrm{T}}(t)]^{\mathrm{T}}\delta\boldsymbol{L}_x^{\mathrm{T}}(t) + \delta\boldsymbol{L}_x(t)[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]\delta\boldsymbol{L}_x^{\mathrm{T}}(t)\\ &= \boldsymbol{M}(t) + \boldsymbol{M}^{\mathrm{T}}(t) + \delta\boldsymbol{L}_x(t)[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]\delta\boldsymbol{L}_x^{\mathrm{T}}(t), \quad (11)\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{M}(t) &:= -\delta\boldsymbol{L}_x(t)[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t)\\ &\quad + \boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{B}^{\mathrm{T}}u(t) - R_v\boldsymbol{L}_x^{\mathrm{T}}(t)]. \quad (12)\end{aligned}$$

If take $\boldsymbol{M}(t) = \boldsymbol{0}$, i.e.,

$$\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t) + \boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{B}^{\mathrm{T}}u(t) - R_v\boldsymbol{L}_x^{\mathrm{T}}(t) = \boldsymbol{0},$$

then we can obtain

$$\boldsymbol{L}_x(t) = [\boldsymbol{A} + \boldsymbol{B}u(t)]\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1}. \quad (13)$$

Thus, we have

$$\delta\boldsymbol{P}_x(t+1) = \delta\boldsymbol{L}_x(t)[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]\delta\boldsymbol{L}_x^{\mathrm{T}}(t). \quad (14)$$

From (14), it can be found that $\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v$ is non-negative at least because R_v is non-negative, the matrix $\boldsymbol{P}_x(t)$ is non-negative definite. If $\delta\boldsymbol{L}_x(t) \neq \boldsymbol{0}$, then $\delta\boldsymbol{P}_x(t+1)$ is the non-negative definite matrix, which explains that the non-negative deviation is generated when any departure $\delta\boldsymbol{L}_x(t)$ effects the filtering gain vector $\boldsymbol{L}_x(t)$. Therefore, $\boldsymbol{L}_x(t)$ in (13) is the optimal gain vector which makes the state estimation error covariance matrix minimum. To summarize, the bilinear state observer is as follows,

$$\begin{aligned}\hat{\boldsymbol{x}}(t+1) &= \boldsymbol{A}\hat{x}(t) + \boldsymbol{B}\hat{x}(t)u(t) + \boldsymbol{f}u(t)\\ &\quad + \boldsymbol{L}_x(t)[y(t) - \boldsymbol{c}\hat{\boldsymbol{x}}(t)], \quad \hat{\boldsymbol{x}}(1) = \boldsymbol{1}_n/p_0, \quad (15)\\ \boldsymbol{L}_x(t) &= \boldsymbol{A}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1}\boldsymbol{B}u(t)\boldsymbol{P}_x(t)\\ &\quad \times\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1}, \quad \boldsymbol{P}_x(1) = \boldsymbol{I}_n, \quad (16)\\ \boldsymbol{P}_x(t+1) &= [\boldsymbol{A} - \boldsymbol{L}_x(t)\boldsymbol{c} + \boldsymbol{B}u(t)]\boldsymbol{P}_x(t)[\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t)\\ &\quad + \boldsymbol{B}^{\mathrm{T}}u(t)] + \boldsymbol{L}_x(t)R_v\boldsymbol{L}_x^{\mathrm{T}}(t). \quad (17)\end{aligned}$$

If the parameter matrices/vector $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{f}$ are unknown, the bilinear state observer in (15)–(17) cannot obtain the state estimates. Then we apply the parameter identification algorithm to give the parameter estimates, and use the estimated parameters to compute the estimates $\hat{\boldsymbol{x}}(t)$ of the states $\boldsymbol{x}(t)$. Define the estimates of the parameter vectors $\boldsymbol{\theta}$, $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{f}$ as

$$\begin{aligned}\hat{\boldsymbol{\theta}}(t) &:= [\hat{\boldsymbol{a}}^{\mathrm{T}}(t), \hat{\boldsymbol{b}}^{\mathrm{T}}(t), \hat{\boldsymbol{f}}^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{n^2+2n},\\ \hat{\boldsymbol{a}}(t) &:= [\hat{a}_1(t), \hat{a}_2(t), \ldots, \hat{a}_n(t)]^{\mathrm{T}} \in \mathbb{R}^n,\end{aligned}$$

$$\hat{\boldsymbol{b}}(t) := [\hat{\boldsymbol{b}}_1(t), \hat{\boldsymbol{b}}_2(t), \ldots, \hat{\boldsymbol{b}}_n(t)]^{\mathrm{T}} \in \mathbb{R}^{n^2},$$
$$\hat{\boldsymbol{f}}(t) := [\hat{f}_1(t), \hat{f}_2(t), \ldots, \hat{f}_n(t)]^{\mathrm{T}} \in \mathbb{R}^{n}.$$

Use $\hat{\boldsymbol{a}}(t)$, $\hat{\boldsymbol{b}}(t)$ and $\hat{\boldsymbol{f}}(t)$ to construct the estimates $\hat{\boldsymbol{A}}(t)$, $\hat{\boldsymbol{B}}(t)$ and $\hat{\boldsymbol{f}}(t)$ of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{f}$ as

$$\hat{\boldsymbol{A}}(t) := \begin{bmatrix} -\hat{a}_1(t) & 1 & 0 & \cdots & 0 \\ -\hat{a}_2(t) & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -\hat{a}_{n-1}(t) & 0 & \cdots & 0 & 1 \\ -\hat{a}_n(t) & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad (18)$$

$$\hat{\boldsymbol{B}}(t) := \begin{bmatrix} \hat{\boldsymbol{b}}_1(t) \\ \hat{\boldsymbol{b}}_2(t) \\ \vdots \\ \hat{\boldsymbol{b}}_n(t) \end{bmatrix}, \quad \hat{\boldsymbol{f}}(t) := \begin{bmatrix} \hat{f}_1(t) \\ \hat{f}_2(t) \\ \vdots \\ \hat{f}_n(t) \end{bmatrix}. \quad (19)$$

Replacing $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{f}$ in (15)–(17) with $\hat{\boldsymbol{A}}(t)$, $\hat{\boldsymbol{B}}(t)$ and $\hat{\boldsymbol{f}}(t)$ gives

$$\hat{\boldsymbol{x}}(t+1) = \hat{\boldsymbol{A}}(t)\hat{x}(t) + \hat{\boldsymbol{B}}(t)\hat{x}(t)u(t) + \hat{\boldsymbol{f}}(t)u(t) + \boldsymbol{L}_x(t)[y(t) - \boldsymbol{c}\hat{\boldsymbol{x}}(t)], \quad \hat{\boldsymbol{x}}(1) = \mathbf{1}_n/p_0, \quad (20)$$
$$\boldsymbol{L}_x(t) = \hat{\boldsymbol{A}}(t)\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1}\hat{\boldsymbol{B}}(t)u(t)\boldsymbol{P}_x(t) \times\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1}, \quad \boldsymbol{P}_x(1) = \boldsymbol{I}_n, \quad (21)$$
$$\boldsymbol{P}_x(t+1) = [\hat{\boldsymbol{A}}(t) - \boldsymbol{L}_x(t)\boldsymbol{c} + \hat{\boldsymbol{B}}(t)u(t)]\boldsymbol{P}_x(t)[\hat{\boldsymbol{A}}^{\mathrm{T}}(t) - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t) + \hat{\boldsymbol{B}}^{\mathrm{T}}(t)u(t)] + \boldsymbol{L}_x(t)R_v\boldsymbol{L}_x^{\mathrm{T}}(t). \quad (22)$$

Equations (18)–(22) form the parameter estimates based bilinear state observer to compute the estimates $\hat{\boldsymbol{x}}(t)$ of the state vector $\boldsymbol{x}(t)$.

4 The BSO based recursive least squares algorithm

In this section, a bilinear state observer based recursive least squares algorithm is employed to solve the combined state and parameter estimation problem of the considered bilinear system.

From (4)–(6), we find that the RLS algorithm cannot obtain the unknown parameters because of the unknown states $\boldsymbol{x}(t)$. So based on the bilinear state observer proposed in Section 3, we replace the unknown state $\boldsymbol{x}(t-i)$ with its estimate $\hat{\boldsymbol{x}}(t-i)$ and define

$$\hat{\boldsymbol{\varphi}}(t) := [\hat{\boldsymbol{\varphi}}_x^{\mathrm{T}}(t), \hat{\boldsymbol{\varphi}}_{xu}^{\mathrm{T}}(t), \boldsymbol{\varphi}_u^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{n^2+2n},$$
$$\hat{\boldsymbol{\varphi}}_x(t) := [-\hat{x}_1(t-1), -\hat{x}_1(t-2), \ldots, -\hat{x}_1(t-n)]^{\mathrm{T}} \in \mathbb{R}^{n},$$
$$\hat{\boldsymbol{\varphi}}_{xu}(t) := [\hat{\boldsymbol{x}}^{\mathrm{T}}(t-1)u(t-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(t-2)u(t-2), \ldots, \hat{\boldsymbol{x}}^{\mathrm{T}}(t-n)u(t-n)]^{\mathrm{T}} \in \mathbb{R}^{n^2}.$$

Moreover, employing the bilinear state observer in (18)–(22), and replacing $\boldsymbol{x}(t)$, $\boldsymbol{\varphi}(t)$, $\boldsymbol{\varphi}_x(t)$ and $\boldsymbol{\varphi}_{xu}(t)$ with their estimates $\hat{\boldsymbol{x}}(t)$, $\hat{\boldsymbol{\varphi}}(t)$, $\hat{\boldsymbol{\varphi}}_x(t)$ and $\hat{\boldsymbol{\varphi}}_{xu}(t)$ in (4)–(6) gives the bilinear state observer based recursive least squares (BSO-RLS) algorithm:

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{L}(t)[y(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)], \quad (23)$$
$$\boldsymbol{L}(t) = \boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)[1 + \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)]^{-1}, \quad (24)$$
$$\boldsymbol{P}(t) = [\boldsymbol{I} - \boldsymbol{L}(t)\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)]\boldsymbol{P}(t-1), \quad (25)$$
$$\hat{\boldsymbol{\varphi}}(t) = [\hat{\boldsymbol{\varphi}}_x^{\mathrm{T}}(t), \hat{\boldsymbol{\varphi}}_{xu}^{\mathrm{T}}(t), \boldsymbol{\varphi}_u^{\mathrm{T}}(t)]^{\mathrm{T}}, \quad (26)$$
$$\hat{\boldsymbol{\varphi}}_x(t) = [-\hat{x}_1(t-1), \ldots, -\hat{x}_1(t-n)]^{\mathrm{T}}, \quad (27)$$
$$\hat{\boldsymbol{\varphi}}_{xu}(t) = [\hat{\boldsymbol{x}}^{\mathrm{T}}(t-1)u(t-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(t-2)u(t-2), \ldots, \hat{\boldsymbol{x}}^{\mathrm{T}}(t-n)u(t-n)]^{\mathrm{T}}, \quad (28)$$
$$\boldsymbol{\varphi}_u(t) = [u(t-1), u(t-2), \ldots, u(t-n)]^{\mathrm{T}}, \quad (29)$$
$$\hat{\boldsymbol{x}}(t+1) = \hat{\boldsymbol{A}}(t)\hat{x}(t) + \hat{\boldsymbol{B}}(t)\hat{x}(t)u(t) + \hat{\boldsymbol{f}}(t)u(t) + \boldsymbol{L}_x(t)[y(t) - \boldsymbol{c}\hat{\boldsymbol{x}}(t)], \quad (30)$$
$$\boldsymbol{L}_x(t) = \hat{\boldsymbol{A}}(t)\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1} + \hat{\boldsymbol{B}}(t)u(t)\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1}, \quad (31)$$
$$\boldsymbol{P}_x(t+1) = [\hat{\boldsymbol{A}}(t) - \boldsymbol{L}_x(t)\boldsymbol{c} + \hat{\boldsymbol{B}}(t)u(t)]\boldsymbol{P}_x(t)[\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t) + \hat{\boldsymbol{B}}^{\mathrm{T}}(t)u(t)] + \boldsymbol{L}_x(t)R_v\boldsymbol{L}_x^{\mathrm{T}}(t), \quad (32)$$
$$\hat{\boldsymbol{A}}(t) = \begin{bmatrix} -\hat{a}_1(t) & 1 & 0 & \cdots & 0 \\ -\hat{a}_2(t) & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -\hat{a}_{n-1}(t) & 0 & \cdots & 0 & 1 \\ -\hat{a}_n(t) & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad (33)$$
$$\hat{\boldsymbol{B}}(t) = \begin{bmatrix} \hat{\boldsymbol{b}}_1(t) \\ \hat{\boldsymbol{b}}_2(t) \\ \vdots \\ \hat{\boldsymbol{b}}_n(t) \end{bmatrix}, \quad \hat{\boldsymbol{f}}(t) = \begin{bmatrix} \hat{f}_1(t) \\ \hat{f}_2(t) \\ \vdots \\ \hat{f}_n(t) \end{bmatrix}, \quad (34)$$
$$\hat{\boldsymbol{\theta}}(t) = [\hat{\boldsymbol{a}}^{\mathrm{T}}(t), \hat{\boldsymbol{b}}^{\mathrm{T}}(t), \hat{\boldsymbol{f}}^{\mathrm{T}}(t)]^{\mathrm{T}}. \quad (35)$$

To summarize, we list the steps for implementing the algorithm in (23)–(35) as follows.

1. To initialize: let $t=1$, $\hat{\boldsymbol{\theta}}(0) = \mathbf{1}_{n^2+2n}/p_0$, $\hat{\boldsymbol{x}}(1) = \mathbf{1}_n/p_0$, $\boldsymbol{P}(0) = p_0\boldsymbol{I}_{n^2+2n}$, $\boldsymbol{P}_x(1) = \boldsymbol{I}_n$, $p_0 = 10^6$.
2. Collect the measurement data $u(t)$ and $y(t)$. Form $\hat{\boldsymbol{\varphi}}(t)$ using (26)–(29).
3. Compute the gain vector $\boldsymbol{L}(t)$ using (24), the covariance matrix $\boldsymbol{P}(t)$ using (25).
4. Update the parameter estimate $\hat{\boldsymbol{\theta}}(t)$ using (23).
5. Read $\hat{a}_i(t)$, $\hat{\boldsymbol{b}}_i(t)$ and $\hat{f}_i(t)$ from $\hat{\boldsymbol{\theta}}(t)$ in (35). Construct $\hat{\boldsymbol{A}}(t)$, $\hat{\boldsymbol{B}}(t)$ and $\hat{\boldsymbol{f}}(t)$ using (33)–(34).
6. Compute $\boldsymbol{L}_x(t)$ and $\boldsymbol{P}_x(t+1)$ using (31)–(32).
7. Compute $\hat{\boldsymbol{x}}(t+1)$ using (30).
8. Increase t by 1 and turn to Step 2.

5 The BSO based hierarchical least squares algorithm

In order to improve the computational efficiency, based on the hierarchical identification principle, we decompose the original system into three fictitious subsystems and derive a hierarchical least squares algorithm to estimate the unknown parameters of the bilinear system.

Define three intermediate variables:

$$y_x(t) := y(t) - \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\boldsymbol{b} - \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\boldsymbol{f} \in \mathbb{R}, \quad (36)$$
$$y_{xu}(t) := y(t) - \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\boldsymbol{a} - \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\boldsymbol{f} \in \mathbb{R}, \quad (37)$$
$$y_u(t) := y(t) - \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\boldsymbol{a} - \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\boldsymbol{b} \in \mathbb{R}. \quad (38)$$

Decompose the system in (1)–(2) into three subsystems:

$$y_x(t) = \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\boldsymbol{a} + v(t), \quad (39)$$
$$y_{xu}(t) = \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\boldsymbol{b} + v(t), \quad (40)$$
$$y_u(t) = \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\boldsymbol{f} + v(t). \quad (41)$$

Let $\hat{\boldsymbol{a}}(t)$, $\hat{\boldsymbol{b}}(t)$ and $\hat{\boldsymbol{f}}(t)$ be the estimates of $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{f}$ at time t. For the sub-identification models in (39)–(41), define and minimize the following three cost functions

$$J(\boldsymbol{a}) := \sum_{j=1}^{t}[y_x(j) - \boldsymbol{\varphi}_x^{\mathrm{T}}(j)\boldsymbol{a}]^2,$$

IET Research Journals, pp. 1–10

$$J(\boldsymbol{b}) := \sum_{j=1}^{t}[y_{xu}(j) - \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(j)\boldsymbol{b}]^2,$$

$$J(\boldsymbol{f}) := \sum_{j=1}^{t}[y_u(j) - \boldsymbol{\varphi}_u^{\mathrm{T}}(j)\boldsymbol{f}]^2.$$

According to the least squares principle, we get the least squares based recursive relations:

$$\begin{aligned}\hat{\boldsymbol{a}}(t) &= \hat{\boldsymbol{a}}(t-1) + \boldsymbol{L}_1(t)[y_x(t) - \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\hat{\boldsymbol{a}}(t-1)],\\ &= \hat{\boldsymbol{a}}(t-1) + \boldsymbol{L}_1(t)[y(t) - \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\boldsymbol{b}\\ &\quad -\boldsymbol{\varphi}_u^{\mathrm{T}}(t)\boldsymbol{f} - \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\hat{\boldsymbol{a}}(t-1)],\ \hat{\boldsymbol{a}}(0) = \mathbf{1}_n/p_0, \end{aligned} \tag{42}$$

$$\boldsymbol{L}_1(t) = \frac{\boldsymbol{P}_1(t-1)\boldsymbol{\varphi}_x(t)}{1 + \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1)\boldsymbol{\varphi}_x(t)}, \tag{43}$$

$$\boldsymbol{P}_1(t) = [\boldsymbol{I} - \boldsymbol{L}_1(t)\boldsymbol{\varphi}_x^{\mathrm{T}}(t)]\boldsymbol{P}_1(t-1), \tag{44}$$

$$\begin{aligned}\hat{\boldsymbol{b}}(t) &= \hat{\boldsymbol{b}}(t-1) + \boldsymbol{L}_2(t)[y_{xu}(t) - \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t-1)],\\ &= \hat{\boldsymbol{b}}(t-1) + \boldsymbol{L}_2(t)[y(t) - \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\boldsymbol{a}\\ &\quad -\boldsymbol{\varphi}_u^{\mathrm{T}}(t)\boldsymbol{f} - \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t-1)],\ \hat{\boldsymbol{b}}(0) = \mathbf{1}_{n^2}/p_0, \end{aligned} \tag{45}$$

$$\boldsymbol{L}_2(t) = \frac{\boldsymbol{P}_2(t-1)\boldsymbol{\varphi}_{xu}(t)}{1 + \boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\boldsymbol{P}_2(t-1)\boldsymbol{\varphi}_{xu}(t)}, \tag{46}$$

$$\boldsymbol{P}_2(t) = [\boldsymbol{I} - \boldsymbol{L}_2(t)\boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)]\boldsymbol{P}_2(t-1), \tag{47}$$

$$\begin{aligned}\hat{\boldsymbol{f}}(t) &= \hat{\boldsymbol{f}}(t-1) + \boldsymbol{L}_3(t)[y_u(t) - \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\hat{\boldsymbol{f}}(t-1)],\\ &= \hat{\boldsymbol{f}}(t-1) + \boldsymbol{L}_3(t)[y(t) - \boldsymbol{\varphi}_x^{\mathrm{T}}(t)\boldsymbol{a}\\ &\quad -\boldsymbol{\varphi}_{xu}^{\mathrm{T}}(t)\boldsymbol{b} - \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\hat{\boldsymbol{f}}(t-1)],\ \hat{\boldsymbol{f}}(0) = \mathbf{1}_n/p_0, \end{aligned} \tag{48}$$

$$\boldsymbol{L}_3(t) = \frac{\boldsymbol{P}_3(t-1)\boldsymbol{\varphi}_u(t)}{1 + \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\boldsymbol{P}_3(t-1)\boldsymbol{\varphi}_u(t)}, \tag{49}$$

$$\boldsymbol{P}_3(t) = [\boldsymbol{I} - \boldsymbol{L}_3(t)\boldsymbol{\varphi}_u^{\mathrm{T}}(t)]\boldsymbol{P}_3(t-1). \tag{50}$$

Remark 3. However, the information vector $\boldsymbol{\varphi}_x(t)$ and $\boldsymbol{\varphi}_{xu}(t)$ contain the unknown states $\boldsymbol{x}(t)$. Equations (42), (45) and (48) contain the unknown parameters $\boldsymbol{f}$, $\boldsymbol{b}$ and $\boldsymbol{a}$, so the algorithm in (42)–(50) cannot compute the parameter estimates. This problem will be solved by a coordination approach.

Remark 4. The strategy of the coordination is to employ the bilinear state observer in (18)–(22) to obtain the state estimates, and then replace the unknown states $\boldsymbol{x}(t)$ with its corresponding estimates $\hat{\boldsymbol{x}}(t)$ and the information vector $\boldsymbol{\varphi}(t)$ with $\hat{\boldsymbol{\varphi}}(t)$, $\boldsymbol{\varphi}_x(t)$ with $\hat{\boldsymbol{\varphi}}_x(t)$ and $\boldsymbol{\varphi}_{xu}(t)$ with $\hat{\boldsymbol{\varphi}}_{xu}(t)$. The unknown parameters $\boldsymbol{f}$, $\boldsymbol{b}$ in (42) are replaced with $\hat{\boldsymbol{f}}(t-1)$ and $\hat{\boldsymbol{b}}(t-1)$. The unknown parameters $\boldsymbol{a}$ and $\boldsymbol{f}$ in (45) are replaced with $\hat{\boldsymbol{a}}(t)$ and $\hat{\boldsymbol{f}}(t-1)$. The unknown parameters $\boldsymbol{a}$ and $\boldsymbol{b}$ in (48) are replaced with $\hat{\boldsymbol{a}}(t)$ and $\hat{\boldsymbol{b}}(t)$.

Equations (15)–(17) contains the system parameters $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{f}$ to be estimated. So the algorithm in (15)–(17) cannot be used directly to estimate the system states $\boldsymbol{x}(t)$. The method is to use the estimated parameters $\hat{\boldsymbol{a}}(t)$, $\hat{\boldsymbol{b}}(t)$ and $\hat{\boldsymbol{f}}(t)$ obtained by the parameter estimation algorithm in (42)–(50) to construct the estimates of the system parameter matrices and parameter vector $\hat{\boldsymbol{A}}(t)$, $\hat{\boldsymbol{B}}(t)$ and $\hat{\boldsymbol{f}}(t)$ to take place of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{f}$. Similarly, replacing $\boldsymbol{x}(t)$ in $\boldsymbol{\varphi}(t)$ with $\hat{\boldsymbol{x}}(t)$ gives the estimates $\hat{\boldsymbol{\varphi}}_x(t)$, $\hat{\boldsymbol{\varphi}}_{xu}(t)$ and $\hat{\boldsymbol{\varphi}}(t)$ in (42)–(50). Then we obtain

$$\begin{aligned}\hat{\boldsymbol{a}}(t) &= \hat{\boldsymbol{a}}(t-1) + \boldsymbol{L}_1(t)[y(t) - \hat{\boldsymbol{\varphi}}_{xu}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t-1)\\ &\quad -\boldsymbol{\varphi}_u^{\mathrm{T}}(t)\hat{\boldsymbol{f}}(t-1) - \hat{\boldsymbol{\varphi}}_x^{\mathrm{T}}(t)\hat{\boldsymbol{a}}(t-1)],\end{aligned} \tag{51}$$

$$\boldsymbol{L}_1(t) = \frac{\boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_x(t)}{1 + \hat{\boldsymbol{\varphi}}_x^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_x(t)}, \tag{52}$$

$$\boldsymbol{P}_1(t) = [\boldsymbol{I} - \boldsymbol{L}_1(t)\hat{\boldsymbol{\varphi}}_x^{\mathrm{T}}(t)]\boldsymbol{P}_1(t-1), \tag{53}$$

$$\begin{aligned}\hat{\boldsymbol{b}}(t) &= \hat{\boldsymbol{b}}(t-1) + \boldsymbol{L}_2(t)[y(t) - \hat{\boldsymbol{\varphi}}_x^{\mathrm{T}}(t)\hat{\boldsymbol{a}}(t)\\ &\quad -\boldsymbol{\varphi}_u^{\mathrm{T}}(t)\hat{\boldsymbol{f}}(t-1) - \hat{\boldsymbol{\varphi}}_{xu}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t-1)],\end{aligned} \tag{54}$$

$$\boldsymbol{L}_2(t) = \frac{\boldsymbol{P}_2(t-1)\hat{\boldsymbol{\varphi}}_{xu}(t)}{1 + \hat{\boldsymbol{\varphi}}_{xu}^{\mathrm{T}}(t)\boldsymbol{P}_2(t-1)\hat{\boldsymbol{\varphi}}_{xu}(t)}, \tag{55}$$

$$\boldsymbol{P}_2(t) = [\boldsymbol{I} - \boldsymbol{L}_2(t)\hat{\boldsymbol{\varphi}}_{xu}^{\mathrm{T}}(t)]\boldsymbol{P}_2(t-1), \tag{56}$$

$$\begin{aligned}\hat{\boldsymbol{f}}(t) &= \hat{\boldsymbol{f}}(t-1) + \boldsymbol{L}_3(t)[y(t) - \hat{\boldsymbol{\varphi}}_x^{\mathrm{T}}(t)\hat{\boldsymbol{a}}(t)\\ &\quad -\hat{\boldsymbol{\varphi}}_{xu}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t) - \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\hat{\boldsymbol{f}}(t-1)],\end{aligned} \tag{57}$$

$$\boldsymbol{L}_3(t) = \frac{\boldsymbol{P}_3(t-1)\boldsymbol{\varphi}_u(t)}{1 + \boldsymbol{\varphi}_u^{\mathrm{T}}(t)\boldsymbol{P}_3(t-1)\boldsymbol{\varphi}_u(t)}, \tag{58}$$

$$\boldsymbol{P}_3(t) = [\boldsymbol{I} - \boldsymbol{L}_3(t)\boldsymbol{\varphi}_u^{\mathrm{T}}(t)]\boldsymbol{P}_3(t-1), \tag{59}$$

$$\hat{\boldsymbol{\varphi}}_x(t) = [-\hat{x}_1(t-1), \ldots, -\hat{x}_1(t-n)]^{\mathrm{T}}, \tag{60}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}_{xu}(t) &= [\hat{\boldsymbol{x}}^{\mathrm{T}}(t-1)u(t-1), \hat{\boldsymbol{x}}^{\mathrm{T}}(t-2)u(t-2), \ldots,\\ &\quad \hat{\boldsymbol{x}}^{\mathrm{T}}(t-n)u(t-n)]^{\mathrm{T}},\end{aligned} \tag{61}$$

$$\boldsymbol{\varphi}_u(t) = [u(t-1), u(t-2), \ldots, u(t-n)]^{\mathrm{T}}, \tag{62}$$

$$\begin{aligned}\hat{\boldsymbol{x}}(t+1) &= \hat{\boldsymbol{A}}(t)\hat{x}(t) + \hat{\boldsymbol{B}}(t)\hat{x}(t)u(t) + \hat{\boldsymbol{f}}(t)u(t)\\ &\quad +\boldsymbol{L}_x(t)[y(t) - \boldsymbol{c}\hat{\boldsymbol{x}}(t)],\end{aligned} \tag{63}$$

$$\begin{aligned}\boldsymbol{L}_x(t) &= \hat{\boldsymbol{A}}(t)\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1}\\ &\quad +\hat{\boldsymbol{B}}(t)u(t)\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}} + R_v]^{-1},\end{aligned} \tag{64}$$

$$\begin{aligned}\boldsymbol{P}_x(t+1) &= [\hat{\boldsymbol{A}}(t) - \boldsymbol{L}_x(t)\boldsymbol{c} + \hat{\boldsymbol{B}}(t)u(t)]\boldsymbol{P}_x(t)[\boldsymbol{A}^{\mathrm{T}}\\ &\quad -\boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t) + \hat{\boldsymbol{B}}^{\mathrm{T}}(t)u(t)] + \boldsymbol{L}_x(t)R_v\boldsymbol{L}_x^{\mathrm{T}}(t),\end{aligned} \tag{65}$$

$$\hat{\boldsymbol{A}}(t) = \begin{bmatrix} -\hat{a}_1(t) & 1 & 0 & \cdots & 0\\ -\hat{a}_2(t) & 0 & 1 & \ddots & 0\\ \vdots & \vdots & \ddots & \ddots & 0\\ -\hat{a}_{n-1}(t) & 0 & \cdots & 0 & 1\\ -\hat{a}_n(t) & 0 & \cdots & 0 & 0 \end{bmatrix}, \tag{66}$$

$$\hat{\boldsymbol{B}}(t) = \begin{bmatrix} \hat{\boldsymbol{b}}_1(t)\\ \hat{\boldsymbol{b}}_2(t)\\ \vdots\\ \hat{\boldsymbol{b}}_n(t) \end{bmatrix},\quad \hat{\boldsymbol{f}}(t) = \begin{bmatrix} \hat{f}_1(t)\\ \hat{f}_2(t)\\ \vdots\\ \hat{f}_n(t) \end{bmatrix}. \tag{67}$$

Equations (51)–(67) form the BSO-HLS algorithm for the bilinear system in (1)–(2), which realize the interactive estimation of bilinear system states and parameters. The steps for implementing the BSO-HLS algorithm as follows.

1. To initialize: let $t = 1$, $\hat{\boldsymbol{a}}(0) = \mathbf{1}_n/p_0$, $\hat{\boldsymbol{b}}(0) = \mathbf{1}_{n^2}/p_0$, $\hat{\boldsymbol{f}}(0) = \mathbf{1}_n/p_0$, $\hat{\boldsymbol{x}}(1) = \mathbf{1}_n/p_0$, $\boldsymbol{P}_1(0) = p_0\boldsymbol{I}_n$, $\boldsymbol{P}_2(0) = p_0\boldsymbol{I}_{n^2}$, $\boldsymbol{P}_3(0) = p_0\boldsymbol{I}_n$, $\boldsymbol{P}_x(1) = \boldsymbol{I}_n$, $p_0 = 10^6$.
2. Collect the measurement data $u(t)$ and $y(t)$. Form $\hat{\boldsymbol{\varphi}}_x(t)$, $\hat{\boldsymbol{\varphi}}_{xu}(t)$ and $\hat{\boldsymbol{\varphi}}_u(t)$ using (60)–(62).
3. Compute the gain vector $\boldsymbol{L}_1(t)$ using (52) and the covariance matrix $\boldsymbol{P}_1(t)$ using (53). Update the parameter estimates $\hat{\boldsymbol{a}}(t)$ using (51).
4. Compute the gain vector $\boldsymbol{L}_2(t)$ using (55) and the covariance matrix $\boldsymbol{P}_2(t)$ using (56). Update the parameter estimates $\hat{\boldsymbol{b}}(t)$ using (54).
5. Compute the gain vector $\boldsymbol{L}_3(t)$ using (58), the covariance matrix $\boldsymbol{P}_3(t)$ using (59). Update the parameter estimates $\hat{\boldsymbol{f}}(t)$ using (57).
6. Construct $\hat{\boldsymbol{A}}(t)$, $\hat{\boldsymbol{B}}(t)$, $\hat{\boldsymbol{f}}(t)$ using (66)–(67).
7. Compute the observer gain vector $\boldsymbol{L}_x(t)$ and $\boldsymbol{P}_x(t+1)$ using (64)–(65). Compute the state estimates $\hat{\boldsymbol{x}}(t+1)$ using (63).
8. Increase t by 1 and turn to Step 2.

In order to improve the BSO-RLS algorithm ability, the next section presents a forgetting factor recursive least squares algorithm based on the bilinear state observer.

6 The BSO-FF-RLS algorithm

To improve the tracking capability of the BSO-RLS algorithm, we introduce a weighting matrix constructed by a forgetting factor in the cost function of the general RLS algorithm.

The identification model of the bilinear system is rewritten as

$$y(t)=\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}+v(t). \tag{68}$$

Define the stacked output vector $\boldsymbol{Y}_t$, the stacked information matrix $\boldsymbol{H}_t$, the stacked error vector $\boldsymbol{V}_t$ as

$$\boldsymbol{Y}_t:=\begin{bmatrix}y(1)\\ y(2)\\ \vdots\\ y(t)\end{bmatrix}\in\mathbb{R}^t,\quad \boldsymbol{H}_t:=\begin{bmatrix}\boldsymbol{\varphi}^{\mathrm{T}}(1)\\ \boldsymbol{\varphi}^{\mathrm{T}}(2)\\ \vdots\\ \boldsymbol{\varphi}^{\mathrm{T}}(t)\end{bmatrix}\in\mathbb{R}^{t\times(n^2+2n)},$$

$$\boldsymbol{V}_t:=\begin{bmatrix}v(1)\\ v(2)\\ \vdots\\ v(t)\end{bmatrix}\in\mathbb{R}^t.$$

Then Equation (68) can be expressed as

$$\boldsymbol{Y}_t=\boldsymbol{H}_t\boldsymbol{\theta}+\boldsymbol{V}_t. \tag{69}$$

Define a quadratic loss function

$$\begin{aligned}J(\hat{\boldsymbol{\theta}}(t))&:=\boldsymbol{V}_t^{\mathrm{T}}\boldsymbol{W}_t\boldsymbol{V}_t\\ &=[\boldsymbol{Y}_t-\boldsymbol{H}_t\hat{\boldsymbol{\theta}}(t)]^{\mathrm{T}}\boldsymbol{W}_t[\boldsymbol{Y}_t-\boldsymbol{H}_t\hat{\boldsymbol{\theta}}(t)],\end{aligned} \tag{70}$$

where $\boldsymbol{W}_t=\mathrm{diag}[\beta^{t-1},\beta^{t-2},\cdots,\beta^0]$ is the weighting matrix. Computing the partial derivative of $J(\boldsymbol{\theta})$ at $\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(t)$:

$$\frac{\partial J(\hat{\boldsymbol{\theta}}(t))}{\partial\hat{\boldsymbol{\theta}}(t)}=-2\boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{W}_t[\boldsymbol{Y}_t-\boldsymbol{H}_t\hat{\boldsymbol{\theta}}(t)]$$

and making $\frac{\partial J(\hat{\boldsymbol{\theta}}(t))}{\partial\hat{\boldsymbol{\theta}}(t)}$ equal zero gives

$$\hat{\boldsymbol{\theta}}(t)=[\boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{W}_t\boldsymbol{H}_t]^{-1}\boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{W}_t\boldsymbol{Y}_t.$$

Defining the covariance matrix $\boldsymbol{P}_4(t):=[\boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{W}_t\boldsymbol{H}_t]^{-1}$, we have

$$\boldsymbol{P}_4^{-1}(t)=\beta\boldsymbol{P}_4^{-1}(t-1)+\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t). \tag{71}$$

Then, the least squares based recursive relation can be given by

$$\hat{\boldsymbol{\theta}}(t)=\hat{\boldsymbol{\theta}}(t-1)+\boldsymbol{L}_4(t)[y(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)], \tag{72}$$

$$\boldsymbol{L}_4(t)=\frac{\boldsymbol{P}_4(t-1)\boldsymbol{\varphi}(t)}{\beta+\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{P}_4(t-1)\boldsymbol{\varphi}(t)}, \tag{73}$$

$$\boldsymbol{P}_4(t)=\frac{1}{\beta}[\boldsymbol{I}-\boldsymbol{L}_4(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)]\boldsymbol{P}_4(t-1), \tag{74}$$

where $\boldsymbol{L}_4(t)$ is the gain vector. From (72)–(74), we find that the information vector $\boldsymbol{\varphi}(t)$ contains the unknown states, so the algorithm cannot be realized. Thus we take use of the bilinear state observer in (15)–(17) to obtain the state estimates $\hat{\boldsymbol{x}}(t)$ of $\boldsymbol{x}(t)$. Replacing $\boldsymbol{\varphi}(t)$ in (72)–(74) with its estimate $\hat{\boldsymbol{\varphi}}(t)$, and combining the bilinear state observer in (18)–(22) gives

$$\hat{\boldsymbol{\theta}}(t)=\hat{\boldsymbol{\theta}}(t-1)+\boldsymbol{L}_4(t)[y(t)-\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)], \tag{75}$$

$$\boldsymbol{L}_4(t)=\frac{\boldsymbol{P}_4(t-1)\hat{\boldsymbol{\varphi}}(t)}{\beta+\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\boldsymbol{P}_4(t-1)\hat{\boldsymbol{\varphi}}(t)}, \tag{76}$$

$$\boldsymbol{P}_4(t)=\frac{1}{\beta}[\boldsymbol{I}-\boldsymbol{L}_4(t)\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)]\boldsymbol{P}_4(t-1), \tag{77}$$

$$\hat{\boldsymbol{\varphi}}_x(t)=[-\hat{x}_1(t-1),\ldots,-\hat{x}_1(t-n)]^{\mathrm{T}}, \tag{78}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}_{xu}(t)=[&\hat{\boldsymbol{x}}^{\mathrm{T}}(t-1)u(t-1),\hat{\boldsymbol{x}}^{\mathrm{T}}(t-2)u(t-2),\ldots,\\ &\hat{\boldsymbol{x}}^{\mathrm{T}}(t-n)u(t-n)]^{\mathrm{T}},\end{aligned} \tag{79}$$

$$\boldsymbol{\varphi}_u(t)=[u(t-1),u(t-2),\ldots,u(t-n)]^{\mathrm{T}}, \tag{80}$$

$$\begin{aligned}\hat{\boldsymbol{x}}(t+1)=&\hat{\boldsymbol{A}}(t)\hat{x}(t)+\hat{\boldsymbol{B}}(t)\hat{x}(t)u(t)+\hat{\boldsymbol{f}}(t)u(t)\\ &+\boldsymbol{L}_x(t)[y(t)-\boldsymbol{c}\hat{\boldsymbol{x}}(t)],\end{aligned} \tag{81}$$

$$\begin{aligned}\boldsymbol{L}_x(t)=&\hat{\boldsymbol{A}}(t)\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}+R_v]^{-1}\\ &+\hat{\boldsymbol{B}}(t)u(t)\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}[\boldsymbol{c}\boldsymbol{P}_x(t)\boldsymbol{c}^{\mathrm{T}}+R_v]^{-1},\end{aligned} \tag{82}$$

$$\begin{aligned}\boldsymbol{P}_x(t+1)=&[\hat{\boldsymbol{A}}(t)-\boldsymbol{L}_x(t)\boldsymbol{c}+\hat{\boldsymbol{B}}(t)u(t)]\boldsymbol{P}_x(t)[\boldsymbol{A}^{\mathrm{T}}\\ &-\boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_x^{\mathrm{T}}(t)+\hat{\boldsymbol{B}}^{\mathrm{T}}(t)u(t)]+\boldsymbol{L}_x(t)R_v\boldsymbol{L}_x^{\mathrm{T}}(t),\end{aligned} \tag{83}$$

$$\hat{\boldsymbol{A}}(t)=\begin{bmatrix}-\hat{a}_1(t) & 1 & 0 & \cdots & 0\\ -\hat{a}_2(t) & 0 & 1 & \ddots & 0\\ \vdots & \vdots & \ddots & \ddots & 0\\ -\hat{a}_{n-1}(t) & 0 & \cdots & 0 & 1\\ -\hat{a}_n(t) & 0 & \cdots & 0 & 0\end{bmatrix}, \tag{84}$$

$$\hat{\boldsymbol{B}}(t)=\begin{bmatrix}\hat{\boldsymbol{b}}_1(t)\\ \hat{\boldsymbol{b}}_2(t)\\ \vdots\\ \hat{\boldsymbol{b}}_n(t)\end{bmatrix},\quad \hat{\boldsymbol{f}}(t)=\begin{bmatrix}\hat{f}_1(t)\\ \hat{f}_2(t)\\ \vdots\\ \hat{f}_n(t)\end{bmatrix}, \tag{85}$$

which forms the bilinear state observer based forgetting factor recursive least squares algorithm.

In general, the number of the multiplication and addition operations is used to show the calculation amount of an algorithm. A division is treated as a multiplication and a substraction is treated as an addition. A multiplication or an addition operation is called a flop (floating point operation). The total flop is treated as the computation cost of an algorithm. The computation cost of the BSO-RLS algorithm and the BSO-HLS algorithm is shown in Table 1.

The calculation amount of the state estimation algorithm is not taken into consideration. The difference of the computation cost between two algorithms is

$$\begin{aligned}N_1-N_2&=4(n^2+2n)^2+6(n^2+2n)-(4n^4+18n^2+20n)\\ &=16n^3+4n^2-8n.\end{aligned} \tag{86}$$

Since $n\geq 1$, the difference N_1-N_2 is positive, which shows that the computation burden of the BSO-RLS algorithm is heavier than that of the BSO-HLS algorithm. That is to say, the hierarchical principle helps reduce the computation cost.

7 Example

Consider the following observer canonical bilinear state space system:

$$\begin{aligned}\boldsymbol{x}(t+1)=&\begin{bmatrix}-0.30 & 1\\ 0.25 & 0\end{bmatrix}\boldsymbol{x}(t)+\begin{bmatrix}0.10 & 0.15\\ 0.30 & 0.20\end{bmatrix}\boldsymbol{x}(t)u(t)\\ &+\begin{bmatrix}1.15\\ 1.56\end{bmatrix}u(t),\\ y(t)=&[1,\ 0]\boldsymbol{x}(t)+v(t).\end{aligned}$$

The parameter vector to be identified is

$$\boldsymbol{\theta}=[a_1,a_2,b_{11},b_{12},b_{21},b_{22},f_1,f_2]^{\mathrm{T}}$$

Table 1 The computation efficiency of the BSO-RLS and BSO-HLS algorithms

Algorithms	Number of multiplications	Number of additions	Total flop
BSO-RLS	$2n^4+8n^3+12n^2+8n$	$2n^4+8n^3+10n^2+4n$	$N_1:=4n^4+16n^3+22n^2+12n$
BSO-HLS	$2n^4+10n^2+12n$	$2n^4+8n^2+8n$	$N_2:=4n^4+18n^2+20n$

$$=[0.30,-0.25,0.10,0.15,0.30,0.20,1.15,1.56]^{\mathrm{T}}. \quad (87)$$

In simulation, the input $\{u(t)\}$ is taken as an uncorrelated persistent excitation signal sequence, and $\{v(t)\}$ as a white noise sequence with zero mean and variance $\sigma^2=0.10^2$. Take the data length $L=3000$ and apply the BSO-RLS and BSO-HLS algorithms to estimate the parameter vector $\boldsymbol{\theta}$ and the states $\boldsymbol{x}(t)$ of the system. The parameter estimates and errors $\delta=\|\hat{\boldsymbol{\theta}}(t)-\boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ are shown in Table 2 and Figure 2, and Table 3 and Figure 5 with $\sigma^2=0.10^2$. The BSO-RLS parameter estimates are shown in Figures 3 and 4.

Based on the BSO-RLS algorithm, choose the forgetting factor $\boldsymbol{\beta}=0.99$. The BSO-FF-RLS parameter estimates and errors are shown in Table 4 and Figure 6. The parameter estimates are shown in Figures 7 and 8. The states $x_i(t)$ and their estimates $\hat{x}_i(t)$ against t are shown in Figures 9 and 10.

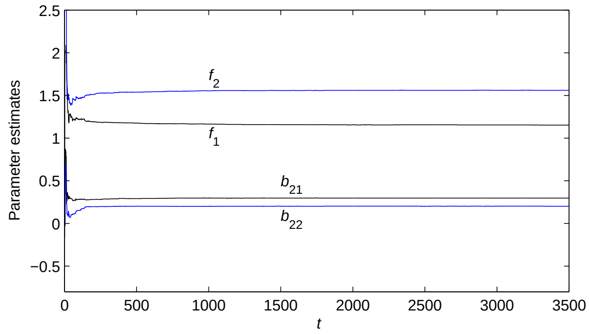

Fig. 4: The BSO-RLS estimates $\hat{b}_{21}(t)$, $\hat{b}_{22}(t)$, $\hat{f}_1(t)$, $\hat{f}_2(t)$ versus t ($\sigma^2=0.10^2$)

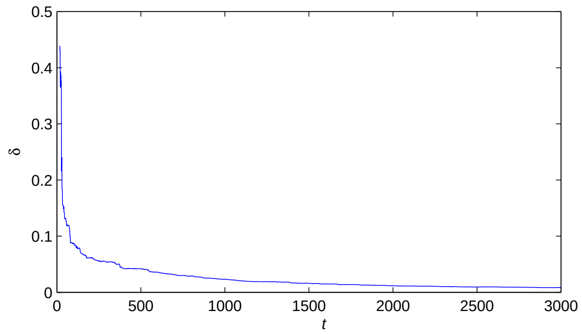

Fig. 2: The BSO-RLS estimation errors δ versus t ($\sigma^2=0.10^2$)

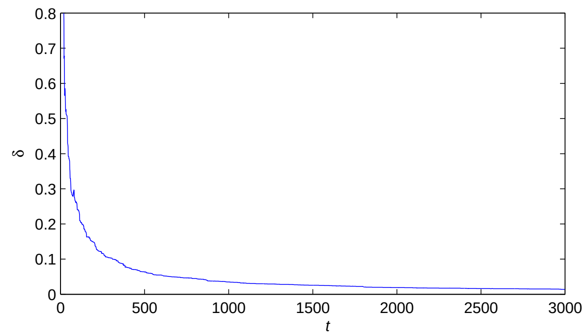

Fig. 5: The BSO-HLS estimation errors δ versus t ($\sigma^2=0.10^2$)

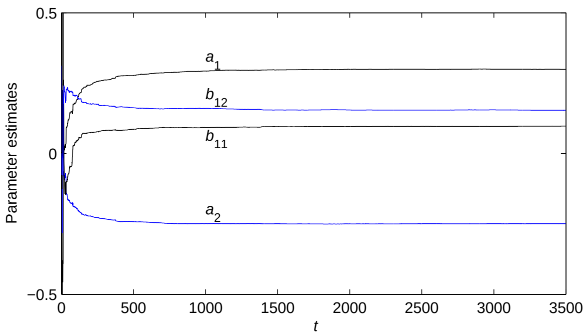

Fig. 3: The BSO-RLS estimates $\hat{a}_1(t)$, $\hat{a}_2(t)$, $\hat{b}_{11}(t)$, $\hat{b}_{12}(t)$ versus t ($\sigma^2=0.10^2$)

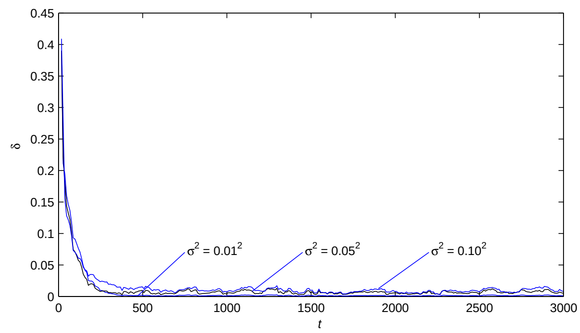

Fig. 6: The BSO-FF-RLS estimation errors δ versus t

From Tables 2–4 and Figures 2–10, we can draw the following conclusions.

1. The parameter estimation errors δ of the proposed BSO-RLS, BSO-HLS and BSO-FF-RLS algorithms become smaller with t increasing, see Tables 2, 3 and 4 and Figures 2, 5 and 6.
2. The parameter estimates reach their true values with the increase of the data length. Moreover, the BSO-RLS algorithm stops the parameter modification after $t>2400$, which means that the parameter accuracy cannot improve even if the recursive process continues, see Figure 3. For comparison, the BSO-FF-RLS parameter estimates continue updating with the data length increasing, which shows that the BSO-FF-RLS algorithm always takes use of the new data information, see Figures 3–4 and 7–8.
3. A small noise variance results in higher accurate parameter estimates, see Figure 6 and Table 4.
4. The state estimates obtained from the bilinear state observer match their true values with t increasing, see Figures 9 and 10.

Table 2 The BSO-RLS estimates and errors ($\sigma^2 = 0.10^2$)

t	a_1	a_2	b_{11}	b_{12}	b_{21}	b_{22}	f_1	f_2	δ (%)
100	0.18252	-0.15717	0.07431	0.17139	0.27482	0.17339	1.15153	1.48225	8.71584
200	0.22467	-0.19104	0.09307	0.15919	0.27461	0.20254	1.15011	1.48691	6.12559
500	0.25421	-0.22152	0.09509	0.15442	0.28348	0.20567	1.15948	1.49865	4.17901
1000	0.27539	-0.23708	0.09752	0.15573	0.29256	0.20197	1.15652	1.52449	2.31081
2000	0.28818	-0.24418	0.09889	0.15230	0.29386	0.20390	1.15066	1.54207	1.16763
3000	0.29181	-0.24600	0.09835	0.15371	0.29527	0.20256	1.15162	1.54752	0.83940
True values	0.30000	-0.25000	0.10000	0.15000	0.30000	0.20000	1.15000	1.56000	

Table 3 The BSO-HLS estimates and errors ($\sigma^2 = 0.10^2$)

t	a_1	a_2	b_{11}	b_{12}	b_{21}	b_{22}	f_1	f_2	δ (%)
100	0.21926	-0.24168	0.19017	-0.06910	0.38898	-0.10189	1.22239	1.82083	24.02848
200	0.29848	-0.27590	0.17698	0.00678	0.35461	-0.00487	1.21911	1.66932	14.76811
500	0.32287	-0.27652	0.12893	0.09151	0.32983	0.11510	1.19150	1.59357	6.35331
1000	0.31685	-0.26185	0.11318	0.12662	0.32310	0.15436	1.17081	1.58774	3.49174
2000	0.31202	-0.25677	0.10738	0.14085	0.31432	0.17827	1.15796	1.57976	1.89333
3000	0.30992	-0.25401	0.10424	0.14625	0.31065	0.18537	1.15662	1.57648	1.39257
True values	0.30000	-0.25000	0.10000	0.15000	0.30000	0.20000	1.15000	1.56000	

Table 4 The BSO-FF-RLS estimates and errors

σ^2	t	a_1	a_2	b_{11}	b_{12}	b_{21}	b_{22}	f_1	f_2	δ (%)
0.10^2	100	0.20899	-0.18206	0.08631	0.16630	0.27579	0.18990	1.13885	1.48345	7.01308
	200	0.26247	-0.22266	0.10104	0.15131	0.28698	0.20906	1.14717	1.51356	3.35240
	500	0.28744	-0.24524	0.10178	0.14576	0.29004	0.20574	1.15964	1.53811	1.49169
	1000	0.30125	-0.24456	0.09294	0.16006	0.30325	0.20390	1.15054	1.56154	0.71927
	2000	0.30668	-0.24770	0.10045	0.14858	0.29435	0.21296	1.15027	1.55866	0.78988
	3000	0.29733	-0.24696	0.09812	0.16167	0.29815	0.20459	1.15884	1.55654	0.81525
0.05^2	100	0.18270	-0.16933	0.08776	0.16369	0.27443	0.18128	1.14551	1.45860	8.85331
	200	0.24951	-0.21963	0.10272	0.14635	0.28043	0.20791	1.14775	1.50008	4.30179
	500	0.29222	-0.24834	0.10101	0.14833	0.29437	0.20305	1.15551	1.54651	0.88712
	1000	0.30060	-0.24730	0.09645	0.15507	0.30163	0.20195	1.15027	1.56072	0.36073
	2000	0.30333	-0.24884	0.10023	0.14928	0.29719	0.20645	1.15013	1.55932	0.39332
	3000	0.29873	-0.24850	0.09905	0.15583	0.29911	0.20229	1.15441	1.55830	0.40619
0.01^2	100	0.15627	-0.15516	0.09072	0.16326	0.27388	0.17251	1.15324	1.43185	10.83426
	200	0.23450	-0.21413	0.10542	0.14185	0.27539	0.20642	1.14906	1.48345	5.46982
	500	0.29555	-0.25079	0.10045	0.15046	0.29781	0.20083	1.15242	1.55253	0.46468
	1000	0.30012	-0.24950	0.09928	0.15103	0.30032	0.20039	1.15005	1.56012	0.07208
	2000	0.30066	-0.24977	0.10005	0.14986	0.29944	0.20129	1.15002	1.55986	0.07840
	3000	0.29975	-0.24970	0.09981	0.15117	0.29983	0.20046	1.15088	1.55967	0.08101
True values		0.30000	-0.25000	0.10000	0.15000	0.30000	0.20000	1.15000	1.56000	

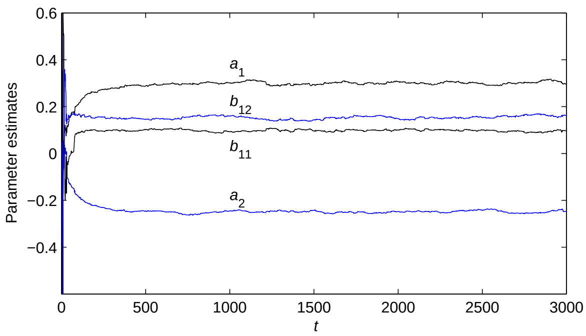

Fig. 7: The BSO-FF-RLS estimates $\hat{a}_1(t)$, $\hat{a}_2(t)$, $\hat{b}_{11}(t)$, $\hat{b}_{12}(t)$ versus t ($\sigma^2 = 0.10^2$)

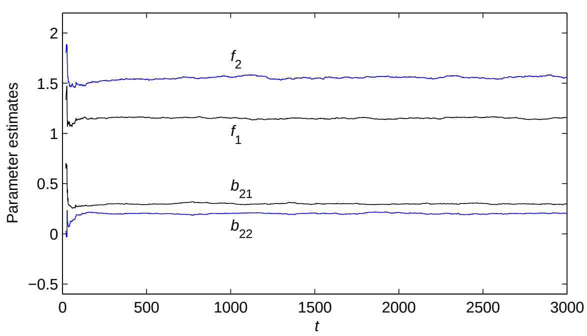

Fig. 8: The BSO-FF-RLS estimates $\hat{b}_{21}(t)$, $\hat{b}_{22}(t)$, $\hat{f}_1(t)$, $\hat{f}_2(t)$ versus t ($\sigma^2 = 0.10^2$)

8 Conclusions

A bilinear state observer based hierarchical least squares algorithm and a bilinear state observer based forgetting factor recursive least squares algorithm are presented for identifying the bilinear systems with unknown states in this paper. The unknown states are obtained under the framework of the bilinear state observer. The decomposition-coordination principle is applied to decompose the bilinear system into three subsystems. The simulation results show

IET Research Journals, pp. 1–10

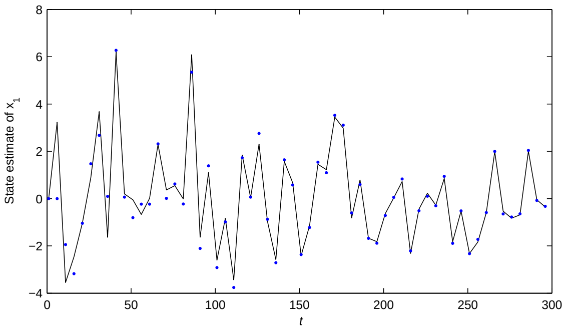

Fig. 9: State $x_1(t)$ and its estimate $\hat{x}_1(t)$ against t ($\sigma^2 = 0.10^2$)

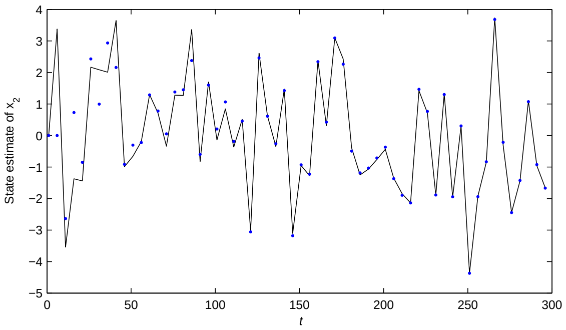

Fig. 10: State $x_2(t)$ and its estimate $\hat{x}_2(t)$ against t ($\sigma^2 = 0.10^2$)

that the proposed algorithms are effective in the parameter identification and state estimation. Compared with the BSO-RLS algorithm, the proposed BSO-HLS algorithm has high computational efficiency. Moreover, the BSO-FF-RLS algorithm has better parameter tracking capability and higher parameter estimate accuracy. The identification method presented in this paper can combine iteration [37, 38] and the data filtering methods to study the identification problems of linear, bilinear and nonlinear systems with different structure and disturbance noise [40–42]. Some mathematical skills [43–48] and statistical methods [49–55] can be used to study the performances of parameter estimation algorithms.

9 Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant no. 61273194) and the 111 Project (grant no. B12018).

10 References

1 Zhang, W., Lin, X., Chen, B.S.: 'LaSalle-type theorem and its applications to infinite horizon optimal control of discrete-time nonlinear stochastic systems", *IEEE Trans. Autom. Control*, 2017, **62**, (1), pp. 250-261

2 Wang, Y., Zhao, D., Li, Y., *et al*.: 'Unbiased minimum variance fault and state estimation for linear discrete time-varying two-dimensional systems', *IEEE Trans. Autom. Control*, 2017, **62**, (10), pp. 5463-5469

3 Zhang, W., Zhao, Y., Sheng, L.: 'Some remarks on stability of stochastic singular systems with state-dependent noise', *Automatica*, 2015, **51**, pp. 273-277

4 Chen, M.T., Ding, F., Xu, L., *et al*.: 'Iterative identification algorithms for bilinear-in-parameter systems with autoregressive moving average noise', *J. Franklin Inst.*, 2017, **354**, (17), pp. 7885-7898

5 Ding, F., Xu, L., Guo, L.J., *et al*.: 'Parameter identification for pseudo-linear systems with ARMA noise using the filtering technique', *IET Control Theory Appl.*, 2018, **12**. doi: 10.1049/iet-cta.2017.0821

6 Xu, L., Ding, F.: 'The parameter estimation algorithms for dynamical response signals based on the multi-innovation theory and the hierarchical principle', *IET Signal Process.*, 2017, **11**, (2) 228-237

7 Ding, F., Wang, Y.J., Dai, J.Y., *et al*.: 'A recursive least squares parameter estimation algorithm for output nonlinear autoregressive systems using the input-output data filtering', *J. Franklin Inst.*, 2017, **354**, (15), pp. 6938-6955

8 Wang, Y., Zhang, H., Wei, S., *et al*.: 'Control performance assessment for ILC-controlled batch processes in two-dimensional system framework', *IEEE Trans. Syst. Man Cybern., Syst.*, 2018, 48, doi: 10.1109/TSMC.2017.2672563

9 Xu, L., Ding, F.: 'Parameter estimation for control systems based on impulse responses', *Int. J. Control Autom. Syst.*, 2017, **15**, (6), pp. 2471-2479

10 Wang, D.Q., Zhang, Z., Yuan, J.Y.: 'Maximum likelihood estimation method for dual-rate Hammerstein systems', *Int. J. Control Autom. Syst.*, 2017, **15**, (2), pp. 698-705

11 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 Process.*, 2018, **147**, pp. 23-34

12 Ma, P., Ding, F., Zhu, Q.M.: 'Decomposition-based recursive least squares identification methods for multivariate pseudolinear systems using the multi-innovation', *Int. J. Syst. Sci.*, 2018, **49**, (5), pp. 920-928

13 Vicario, F., Phan, M.Q., Betti, R., *et al*.: 'Linear state representations for identification of bilinear discrete-time models by interaction matrices', *Nonlinear Dyn.*, 2014, **77**, (4), pp. 1561-1576

14 dos Santos, P.L., Ramos, J.A., de Carvalho, J.L.M.: 'Identification of bilinear systems with white noise inputs: an iterative deterministic-stochastic subspace approach', *IEEE Trans. Control Syst. Technol.*, 2009, **17**, (5), pp. 1145-1153

15 Larkowski, T., Linden, J.G., Vinsonneau, B., *et al*.: 'Frisch scheme identification for dynamic diagonal bilinear models', *Int. J. Control*, 2009, **82**, (9): 1591-1604

16 Li, M.H., Liu, X.M., *et al*.: 'Least-squares-based iterative and gradient-based iterative estimation algorithms for bilinear systems', *Nonlinear Dyn.*, 2017, **89**, (1), pp. 197-211

17 Li, M.H., Liu, X.M., *et al*.: 'The maximum likelihood least squares based iterative estimation algorithm for bilinear systems with autoregressive moving average noise', *J. Franklin Inst.*, 2017, **354**, (12), pp. 4861-4881

18 Li, M.H., Liu, X.M., *et al*.: 'The gradient based iterative estimation algorithms for bilinear systems with autoregressive noise', *Circuits Syst. Signal Process.*, 2017, **36**, (11), pp. 4541-4568

19 Mateos, G., Giannakis, G.B.: 'Distributed recursive least-squares: Stability and performance analysis', *IEEE Trans. Signal Process.*, 2012, **60**, (7), pp. 3740-3754

20 Paleologu, C., Benesty, J., Ciochina, S.: 'A robust variable forgetting factor recursive least-squares algorithm for system identification', *IEEE Signal Process. Lett.*, 2008, **15**, pp. 597-600

21 Wang, D.Q., Mao, L., *et al*.: 'Recasted models based hierarchical extended stochastic gradient method for MIMO nonlinear systems', *IET Control Theory Appl.*, 2017, **11**, (4), pp. 476-485

22 Ma, J.X., Xiong, W.L., Chen, J., *et al*.: 'Hierarchical identification for multivariate Hammerstein systems by using the modified Kalman filter', *IET Control Theory Appl.*, 2017, **11**, (6), pp. 857-869

23 Ding, F., Wang, X.H.: 'Hierarchical stochastic gradient algorithm and its performance analysis for a class of bilinear-in-parameter systems', *Circuits Syst. Signal Process.*, 2017, **36**, (4), pp. 1393-1405

24 Chu, Y.J., Mak, C.M.: 'A variable forgetting factor diffusion recursive least squares algorithm for distributed estimation', *Signal Process.*, 2017, **140**, pp. 219-225

25 Ding, F., Wang, F.F., Xu, L., *et al*.: 'Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique', *IET Control Theory Appl.*, 2017, **11**, (3), pp. 390-400

26 Gong, P.C., Wang, W.Q., Li, F.C., Cheung, H.: 'Sparsity-aware transmit beamspace design for FDA-MIMO radar', *Signal Process.*, 2018, **144**, pp. 99-103.

27 Zhao, N., Chen, Y., Liu, R., Wu, M.H., Xiong, W.: 'Monitoring strategy for relay incentive mechanism in cooperative communication networks', *Computers & Electrical Engineering*, 2017, **60**, pp. 14-29.

28 Zhang, Y.Z., Cao, Y., Wen, Y.H., Liang, L., Zou, F.: 'Optimization of information interaction protocols in cooperative vehicle-infrastructure systems,' *Chinese Journal of Electronics*, 2018, **27**, (2), pp. 439-444.

29 Li, P., Li. R.X., Cao, Y., Xie, G.: 'Multi-objective sizing optimization for island microgrids using triangular aggregation model and Levy-Harmony algorithm', *IEEE Transactions on Industrial Informatics*, 2018. doi: 10.1109/TII.2017.2778079

30 Li, P., Dargaville, R., Cao, Y., et al. Storage aided system property enhancing and hybrid robust smoothing for large-scale PV systems', *IEEE Transactions on Smart Grid*, 2017, **8**, (6), pp. 2871-2879

31 Cao, Y., Ma, L.C., Xiao, S., *et al*.: Standard analysis for transfer delay in CTCS-3', *Chinese Journal of Electronics*, 2017, **26**, (5), pp. 1057-1063

32 Favoreel, W., De Moor, B., Van Overschee, P.: 'Subspace identification of bilinear systems subject to white inputs', *IEEE Trans. Autom. Control*, 1999, **44**, (6), pp. 1157-1165

33 Basin, M., Shi, P., Calderon-Alvarez, D.: 'Joint state filtering and parameter estimation for linear stochastic time-delay systems', *Signal Process.*, 2011, **91**, (4), 782-792

34 Ding, F., Wang, X.H., Mao, L., *et al*.: 'Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering', *Digit. Signal Process.*, 2017, **62**, pp. 211-223

35 Zhang, X., Ding, F., Alsaadi, F.E., *et al*.: 'Recursive parameter identification of the dynamical models for bilinear state space systems', *Nonlinear Dyn.*, 2017, **89**, (4), pp. 2415-2429

36 Zhang, X., Xu, L., *et al*.: 'Combined state and parameter estimation for a bilinear state space system with moving average noise', *J. Franklin Inst.*, 2018, **355**. https://doi.org/10.1016/j.jfranklin.2018.01.011

37 Ding, J.L.: 'Recursive and iterative least squares parameter estimation algorithms for multiple-input-output-error systems with autoregressive noise', *Circuits Syst. Signal Process.*, 2018, **37**. doi: 10.1007/s00034-017-0636-0

38 Ding, F., Meng, D.D., Gai, J.Y., *et al*.: 'Least squares based iterative parameter estimation identification for stochastic dynamical systems with ARMA noise using the model equivalence', *Int. J. Control, Autom. Syst.*, 2018, **16**, (2). http://dx.doi.org/10.1007/s12555-017-0001-x

39 Ding F, Chen HB, *et al*.: 'A hierarchical least squares identification algorithm for Hammerstein nonlinear systems using the key term separation', *J. Franklin Inst.*, 2018, **355**. doi: 10.1016/j.jfranklin.2018.01.052
40 Xu, L., Ding, F.: 'Recursive least squares and multi-innovation stochastic gradient parameter estimation methods for signal modeling', *Circuits Syst. Signal Process.*, 2017, **36**, (4), pp. 1735-1753
41 Xu., L.: 'The parameter estimation algorithms based on the dynamical response measurement data', *Advances in Mechanical Engineering*, 2017, **9**, (11), pp. 1-12. doi: 10.1177/1687814017730003
42 Xu, L., Ding, F.: 'Iterative parameter estimation for signal models based on measured data', *Circuits Syst. Signal Process.*, 2018, **37**. doi: 10.1007/s00034-017-0705-4
43 Liu, F., Wu, H.X.: Singular integrals related to homogeneous mappings in triebel-lizorkin spaces', *Journal of Mathematical Inequalities*, 2017, **11**, (4), pp. 1075-1097
44 Liu, F.: 'Continuity and approximate differentiability of multisublinear fractional maximal functions', *Mathematical Inequalities & Applications*, 2018, **21**, (1), pp. 25-40
45 Liu, F., Wu, H.X.: 'Regularity of discrete multisublinear fractional maximal functions', *Science China – Mathematics*, 2017, **60**, (8), pp. 1461-1476
46 Liu, F., Wu, H.X.: 'On the regularity of maximal operators supported by submanifolds', *Journal of Mathematical Analysis and Applications*, 2017, **453**, (1), pp. 144-158
47 Liu, F.: 'On the Triebel-Lizorkin space boundedness of Marcinkiewicz integrals along compound surfaces', *Mathematical Inequalities & Applications*, 2017, **20**(2), pp. 515-535
48 Liu, F., Xue, Q.Y., Yabuta, K.: 'Rough maximal singular integral and maximal operators supported by subvarieties on Triebel-Lizorkin spaces', *Nonlinear Analysis*, 2018, **171**, pp. 41-72
49 H.L. Gao, C.C. Yin, The perturbed sparre Andersen model with a threshold dividend strategy', *J. Comput. Appl. Math.*, 2008, **220**, (1-2), pp. 394-408
50 Yin, C.C., Wen, Y.Z.: 'Exit problems for jump processes with applications to dividend problems', *J. Comput. Appl. Math.*, 2013, **245**, pp. 30-52
51 Yin, C.C., Wen, Y.Z.: 'Optimal dividend problem with a terminal value for spectrally positive Levy processes', *Insurance Mathematics & Economics*, 2013, **53**, (3), pp. 769-773
52 Yin, C.C., Wang, C.W.: 'The perturbed compound Poisson risk process with investment and debit interest', *Methodology and Computing in Applied Probability*, 2010, **12**, (3), pp. 391-413
53 Yin, C.C., Yuen, K.C.: 'Optimality of the threshold dividend strategy for the compound Poisson model,' *Statistics & Probability Letters*, 2011, **81**, (12), pp. 1841-1846
54 Yin, C.C., Zhao, J.S.: 'Nonexponential asymptotics for the solutions of renewal equations, with applications,' *Journal of Applied Probability*, 2006, **43**, (3), pp. 815-824
55 Ji, Y., Ding, F.: 'Multiperiodicity and exponential attractivity of neural networks with mixed delays', *Circuits Syst. Signal Process.*, 2017, **36**, (6), pp. 2558-2573