

# Highly computationally efficient state filter based on the delta operator

Xiao Zhang<sup>1</sup>, Feng Ding<sup>\*1,2</sup>, Ling Xu<sup>1</sup> and Erfu Yang<sup>3</sup>

*1 Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, PR China*

*2 College of Automation and Electronic Engineering, Qingdao University of Science and Technology, Qingdao 266061, PR China*

*3 Space Mechatronic Systems Technology Laboratory, University of Strathclyde, Glasgow G1 1XJ, Scotland, United Kingdom*

## SUMMARY

The Kalman filter is not suitable for the state estimation of linear systems with multi-state-delays and the extended state vector Kalman filtering algorithm results in heavy computational burden because of the large dimension of the state estimation covariance matrix. Thus, in this paper, we develop a novel state estimation algorithm for enhancing the computational efficiency based on the delta operator. The computation analysis and the simulation example show the performance of the proposed algorithm. Copyright © 2018 John Wiley & Sons, Ltd.

Received ...

**KEY WORDS:** Time-delay system, Kalman filtering, state estimation, delta operator

## 1. INTRODUCTION

As the basis of modern control theory, the state-space representation is an effective mathematical model to totally describe the dynamic behaviors of physical systems [1]. Compared with the transfer function representation [2, 3, 4], it can be applied to more complex systems such as multi-input multi-output systems [5] and nonlinear systems [6]. Filtering methods have been widely used in parameter estimation [7, 8], and the combined state and parameter estimation of state-space systems have attracted much attention throughout the world [9]. Schön studied an expectation maximization algorithm for nonlinear systems described by state-space models and acquired the state estimates through a particle smoother [10]. Partovibakhsh and Liu proposed an adaptive unscented Kalman filtering based approach for jointly online estimation of state-of-charge and parameters of lithium-ion batteries for autonomous mobile robots and computed the noise covariances in the state estimation process by covariance matching [11].

Time delays often exist in signal transmission and signal modeling [12–16] and control systems [17, 18, 19]. For example, in communication, the measurements are often obtained with time delay because of the transmission congestion; the communication networks between subsystems are often unreliable, which will introduce the communication delays. Some important variables of chemical processes are often obtained through online analyzers, resulting large time delays [20]. Such delays may cause instability and poor performance of system dynamics [21]. Thus the analysis and control

\*Correspondence to: Key Laboratory of Advanced Process Control for Light Industry, School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China.

†Email: fding@jiangnan.edu.cn. <http://orcid.org/0000-0002-2721-2025>

of time-delay systems are important [22]. In the literature, Chen et al. utilized a biased compensation recursive least squares algorithm to estimate the parameters and delays of a time-delay rational model [23]. Shi et al. considered the state estimation of Markovian jump neural networks with time-delays, and developed a state estimator to obtain the network state estimates for the error dynamics to be stochastic finite-time stable [24]. Xu utilized the first-order Taylor expansion to approximate the time delay and presented a proportional differential control algorithm for parameter estimation of first-order time-delay model with transfer function [25]. Gu et al. presented an iterative based identification algorithm for linear models with multi-state-delays based on the negative gradient search and the least squares principle, but without considering the state estimation of the unknown states [26]. To the best of our knowledge, the related work has not been reported in the literature on the state estimation of dynamic systems with multi-state-delays. Thus, there is a strong incentive for us to develop an efficient state filter of such systems.

Parameter estimation and state filtering are basic for system identification [27, 28, 29] and system analysis and design [30, 31, 32, 33], and can be applied to many areas [34-39]. The Kalman filter (KF) is known as the optimal state filter for linear systems under the Gaussian white noise and has been extended to study the parameter estimation for bilinear systems [40, 41, 42]. For nonlinear systems, its modifications such as the particle filter, the extended Kalman filter, the unscented Kalman filter give approaches for nonlinear filtering problems. Differing from the above state filters in the recursive way, Zhao et al. developed an unbiased finite impulse response filter to iteratively estimate the state variables using a fixed number of recent measurements [43]. Shi et al. presented a Kalman filter based parameter estimation algorithm on the basis of an output estimator for networked control systems with missing output data and designed an adaptive controller to achieve the output tracking [44].

The KF is employed to provide solutions for the state estimation of the linear system with multi-state-delays. However, according to the computational analysis (see in Section 3), the dimension of the state estimation covariance matrix is large, which causes the high computational burden for state estimation. This motivates us to search for a highly-computationally-efficient state estimation algorithm with high computational efficiency for estimating the system state based on the knowledge of system dynamics and noisy observation data. **The main contributions of this paper are as follows.**

- Study a highly-computationally-efficient state filter for a multi-time-delay system described by a state-space model.
- Apply the delta operator to minimize the state estimation error covariance matrix, resulting in the improvement of computational efficiency.
- Analyze the computational complexity between the presented algorithms to demonstrate the performance of the state filter based on the floating point operation.

The paper is organized as follows. Section 2 formulates the problem and presents a generalized state estimation algorithm by gathering the sub-vectors into an extended state vector. Then a direct state estimation algorithm based on the delta operator is presented to reduce the computational burden in Section 3. The benefits of the proposed methods are shown by the simulation examples in Section 4. Finally, Section 5 provides some concluding remarks.

## 2. THE PROBLEM FORMULATION

Some necessary symbols are introduced as follows.

<b>Symbols</b>	<b>Meaning</b>
$A =: X$	$X$ is defined by $A$ .
$X := A$	$X$ is defined by $A$ .
$\mathbf{1}_n$	An $n$ -dimensional column vector whose entries are all 1.
$E[x]$	The expectation of $x$ .
$\mathbf{I}$	An identity matrix of appropriate size ( $n \times n$ ).

$z$	A unit forward shift operator like $z\mathbf{x}(s) = \mathbf{x}(s+1)$ and $z^{-1}\mathbf{x}(s) = \mathbf{x}(s-1)$ .
$T$	The vector/matrix transpose.

For the state-space system with a unit time-delay, there are some methods for discussing its stability and identification problems [45, 46]. For more general conditions, this paper considers a linear system with multi-time-delays:

$$\begin{aligned} \mathbf{x}(s+1) &= \mathbf{A}\mathbf{x}(s) + \mathbf{B}_1\mathbf{x}(s-1) + \mathbf{B}_2\mathbf{x}(s-2) + \cdots \\ &\quad + \mathbf{B}_r\mathbf{x}(s-r) + \mathbf{b}u(s) + \mathbf{w}(s), \end{aligned} \quad (1)$$

$$y(s) = \mathbf{c}\mathbf{x}(s) + v(s), \quad (2)$$

where  $u(s) \in \mathbb{R}$  denotes the sampled input,  $y(s) \in \mathbb{R}$  is the system output disturbed by a stochastic noise  $v(s)$ ,  $\mathbf{x}(s) \in \mathbb{R}^n$  is the state vector,  $\mathbf{w}(s) \in \mathbb{R}^n$  is the process noise,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B}_q \in \mathbb{R}^{n \times n}$  ( $q = 1, 2, \dots, r$ ),  $\mathbf{b} \in \mathbb{R}^n$  and  $\mathbf{c} \in \mathbb{R}^{1 \times n}$  are the system parameters.

Assume that the system output variable is disturbed by a stochastic noise  $v(s)$  with zero mean and variance  $R_v$ ,  $\mathbf{w}(s) \in \mathbb{R}^n$  is the uncorrected process noise vector with zero mean and variance  $\mathbf{R}_w$ . The noises  $\mathbf{w}(s)$  and  $v(s)$  are uncorrelated and their covariance matrices satisfy

$$(A1) \quad \mathbb{E}[\mathbf{w}(s)] = \mathbf{0}, \quad \mathbb{E}[v(s)] = 0, \quad \mathbb{E}[\mathbf{w}(s)v(i)] = \mathbf{0},$$

$$(A2) \quad \mathbb{E}[\mathbf{w}(s)\mathbf{w}^T(s)] = \mathbf{0}, \quad \mathbb{E}[v(l)v(t)] = 0, \quad l \neq t,$$

$$(A3) \quad \mathbb{E}[\mathbf{w}(s)\mathbf{w}^T(s)] = \mathbf{R}_w \in \mathbb{R}^{n \times n}, \quad \mathbb{E}[v^2(s)] = R_v \in \mathbb{R}.$$

Assume that the system in (1)–(2) is stable, observable and controllable. For a linear dynamic system with multi-state-delays, the Kalman filter cannot be employed directly to obtain the optimal state estimates. Here we present a generalized state estimation algorithm on the basis of the Kalman filter to compute the estimates of the unknown states.

Define the extended state vector  $\mathbf{X}(s) := [\mathbf{x}^T(s), \mathbf{x}^T(s-1), \mathbf{x}^T(s-2), \dots, \mathbf{x}^T(s-r)]^T \in \mathbb{R}^{nr+n}$  and the extended noise vector  $\mathbf{W}(s) := [\mathbf{w}^T(s), 0, \dots, 0]^T \in \mathbb{R}^{nr+n}$ . Then System (1)–(2) can be expressed as

$$\mathbf{X}(s+1) = \mathbf{G}\mathbf{X}(s) + \mathbf{H}u(s) + \mathbf{W}(s), \quad (3)$$

$$y(s) = \mathbf{\Gamma}\mathbf{X}(s) + v(s), \quad (4)$$

where the parameter matrices and vectors are defined as

$$\mathbf{G} := \begin{bmatrix} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_r \\ \mathbf{I}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n & \mathbf{0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}_n & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(nr+n) \times (nr+n)},$$

$$\mathbf{H} := [\mathbf{b}^T, \mathbf{0}, \mathbf{0}, \dots, \mathbf{0}]^T \in \mathbb{R}^{nr+n},$$

$$\mathbf{\Gamma} := [\mathbf{c}, \mathbf{0}, \mathbf{0}, \dots, \mathbf{0}] \in \mathbb{R}^{1 \times (nr+n)}.$$

For the state vector  $\mathbf{X}(s)$ , it is well-known that the Kalman filter can be employed to estimate the unknown states for the linear systems from observation data. Let  $\hat{\mathbf{X}}(s|s-1) \in \mathbb{R}^{nr+n}$  and  $\hat{\mathbf{X}}(s|s) \in \mathbb{R}^{nr+n}$  be the predicted state estimate and the posterior state estimate of the unknown  $\mathbf{X}(s)$ . Let  $\mathbf{K}(s) \in \mathbb{R}^{nr+n}$  denote the optimal gain vector,  $\mathbf{P}(s|s-1) \in \mathbb{R}^{(nr+n) \times (nr+n)}$  denote the priori state estimation error covariance matrix of  $\mathbf{X}(s)$  and  $\mathbf{P}(s|s) \in \mathbb{R}^{(nr+n) \times (nr+n)}$  denote the posteriori state estimation error covariance matrix of  $\mathbf{X}(s)$ . Based on the Kalman filtering principle, we obtain the generalized state estimation algorithm:

$$\hat{\mathbf{X}}(s|s) = \hat{\mathbf{X}}(s|s-1) + \mathbf{K}(s)[y(s) - \mathbf{\Gamma}\hat{\mathbf{X}}(s|s-1)], \quad (5)$$

$$\mathbf{K}(s) = \mathbf{P}(s|s-1)\mathbf{\Gamma}^T[\mathbf{\Gamma}\mathbf{P}(s|s-1)\mathbf{\Gamma}^T + R_v]^{-1}, \quad (6)$$

$$\mathbf{P}(s|s) = [\mathbf{I} - \mathbf{K}(s)\mathbf{\Gamma}]\mathbf{P}(s|s-1)[\mathbf{I} - \mathbf{K}(s)\mathbf{\Gamma}]^T + \mathbf{K}(s)\mathbf{R}_v\mathbf{K}^T(s), \quad (7)$$

$$\hat{\mathbf{X}}(s|s-1) = \mathbf{G}\hat{\mathbf{X}}(s-1|s-1) + \mathbf{H}u(s-1), \quad (8)$$

$$\mathbf{P}(s|s-1) = \mathbf{G}\mathbf{P}(s-1|s-1)\mathbf{G}^T + \mathbf{R}_W. \quad (9)$$

Define the state estimate  $\hat{\mathbf{X}}(s) := \hat{\mathbf{X}}(s|s)$  of  $\mathbf{X}(s)$  at time  $s$  and the covariance matrix  $\mathbf{P}(s) := \mathbf{P}(s|s)$  at time  $s$ . Then through eliminating the intermediate variables  $\mathbf{X}(s|s-1)$  and  $\mathbf{P}(s|s-1)$  in (5) to (9), we obtain the generalized Kalman filtering algorithm:

$$\hat{\mathbf{X}}(s) = \mathbf{G}\hat{\mathbf{X}}(s-1) + \mathbf{H}u(s-1) + \mathbf{K}(s)\{y(s) - \mathbf{\Gamma}[\mathbf{G}\hat{\mathbf{X}}(s-1) + \mathbf{H}u(s-1)]\}, \quad (10)$$

$$\mathbf{K}(s) = [\mathbf{G}\mathbf{P}(s-1)\mathbf{G}^T + \mathbf{R}_W]\mathbf{\Gamma}^T\{\mathbf{\Gamma}[\mathbf{G}\mathbf{P}(s-1)\mathbf{G}^T + \mathbf{R}_W]\mathbf{\Gamma}^T + \mathbf{R}_v\}^{-1}, \quad (11)$$

$$\mathbf{P}(s) = [\mathbf{I} - \mathbf{K}(s)\mathbf{\Gamma}][\mathbf{G}\mathbf{P}(s-1)\mathbf{G}^T + \mathbf{R}_W][\mathbf{I} - \mathbf{K}(s)\mathbf{\Gamma}]^T + \mathbf{K}(s)\mathbf{R}_v\mathbf{K}^T(s). \quad (12)$$

**Remark 1.** Although the state estimation algorithm in (10)–(12) can be directly derived based on the Kalman filtering principle, the most remarkable problem is its heavy computational burden, especially for large-scale systems.

**Remark 2.** Because of the existence of the multi-state-delays, the dimension of the state-space model sharply increases, which makes the dimension of the extended state vector become quite large and causes heavy computational burden. This motivates us to design a highly-efficient state filter for state estimation of the dynamics system with multi-state-delays from noisy observation data.

The objective of this paper is to develop a highly-efficient state estimation algorithm based on the delta operator. This method avoids gathering the sub-state vectors into an extended state vector  $\mathbf{X}(s)$  and greatly improves the computational efficiency – see Table III.

### 3. THE DIRECT STATE ESTIMATION ALGORITHM BASED ON THE DELTA OPERATOR

In order to improve the computational efficiency, this section presents a direct state estimation algorithm in a two-step process for the linear system in (1)–(2).

Let  $\hat{\mathbf{x}}_o(s+1) := \hat{\mathbf{x}}(s+1|s)$  denote the predicted state estimate of  $\mathbf{x}(s+1)$  based on the observation data up to time  $s+1$ , and  $\hat{\mathbf{x}}(s+1) := \hat{\mathbf{x}}(s+1|s+1)$  denote the posteriori state estimate of  $\mathbf{x}(s+1)$  based on the observation data up to and including time  $s+1$ .

**Prediction Step:** The predicted state, i.e., the open-loop state estimate can be expressed as

$$\hat{\mathbf{x}}_o(s+1) = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}_1\hat{\mathbf{x}}(s-1) + \mathbf{B}_2\hat{\mathbf{x}}(s-2) + \cdots + \mathbf{B}_r\hat{\mathbf{x}}(s-r) + \mathbf{b}u(s). \quad (13)$$

Define the priori state error covariance matrix as

$$\mathbf{P}_o(s) := \mathbb{E}\{[\mathbf{x}(s) - \hat{\mathbf{x}}_o(s)][\mathbf{x}(s) - \hat{\mathbf{x}}_o(s)]^T\} \in \mathbb{R}^{n \times n}.$$

Then  $\mathbf{P}_o(s+1)$  can be computed by

$$\begin{aligned} \mathbf{P}_o(s+1) &= \mathbb{E}\{[\mathbf{x}(s+1) - \hat{\mathbf{x}}_o(s+1)][\mathbf{x}(s+1) - \hat{\mathbf{x}}_o(s+1)]^T\} \\ &= \mathbb{E}\{[\mathbf{A}\mathbf{x}(s) + \mathbf{B}_1\mathbf{x}(s-1) + \cdots + \mathbf{B}_r\mathbf{x}(s-r) \\ &\quad + \mathbf{b}u(s) + \mathbf{w}(s) - \mathbf{A}\hat{\mathbf{x}}(s) - \mathbf{B}_1\hat{\mathbf{x}}(s-1) \\ &\quad - \mathbf{B}_2\hat{\mathbf{x}}(s-2) - \cdots - \mathbf{B}_r\hat{\mathbf{x}}(s-r) - \mathbf{b}u(s)][\mathbf{A}\mathbf{x}(s) \\ &\quad + \mathbf{B}_1\mathbf{x}(s-1) + \mathbf{B}_2\mathbf{x}(s-2) + \cdots + \mathbf{B}_r\mathbf{x}(s-r) \\ &\quad + \mathbf{b}u(s) + \mathbf{w}(s) - \mathbf{A}\hat{\mathbf{x}}(s) - \mathbf{B}_1\hat{\mathbf{x}}(s-1) \\ &\quad - \mathbf{B}_2\hat{\mathbf{x}}(s-2) - \cdots - \mathbf{B}_r\hat{\mathbf{x}}(s-r) - \mathbf{b}u(s)]^T\} \\ &= \mathbb{E}\{[\mathbf{A}\mathbf{e}(s) + \mathbf{B}_1\mathbf{e}(s-1) + \mathbf{B}_2\mathbf{e}(s-2) + \cdots \\ &\quad + \mathbf{B}_r\mathbf{e}(s-r) + \mathbf{w}(s)][\mathbf{A}\mathbf{e}(s) + \mathbf{B}_1\mathbf{e}(s-1) \\ &\quad + \mathbf{B}_2\mathbf{e}(s-2) + \cdots + \mathbf{B}_r\mathbf{e}(s-r) + \mathbf{w}(s)]^T\}. \end{aligned} \quad (14)$$

**Update Step:** Once the new measurement data  $u(s+1)$  and  $y(s+1)$  are sampled, the modified state estimate, i.e., the closed-loop state estimate can be expressed as

$$\begin{aligned}\hat{\boldsymbol{x}}(s+1) &= \hat{\boldsymbol{x}}_o(s+1) + \boldsymbol{K}(s+1)[y(s+1) - \boldsymbol{c}\hat{\boldsymbol{x}}_o(s+1)] \\ &= \boldsymbol{A}\hat{\boldsymbol{x}}(s) + \boldsymbol{B}_1\hat{\boldsymbol{x}}(s-1) + \boldsymbol{B}_2\hat{\boldsymbol{x}}(s-2) + \cdots \\ &\quad + \boldsymbol{B}_r\hat{\boldsymbol{x}}(s-r) + \boldsymbol{b}u(s) + \boldsymbol{K}(s+1) \\ &\quad \times [\boldsymbol{c}\boldsymbol{x}(s+1) + v(s+1) - \boldsymbol{c}\hat{\boldsymbol{x}}_o(s+1)],\end{aligned}\tag{15}$$

where  $\boldsymbol{K}(s+1)$  is the state gain vector.

**Remark 3:** The choice of  $\boldsymbol{K}(s+1)$  determines the weight of the predicted state  $\hat{\boldsymbol{x}}_o(s+1)$  and the practical measurement data  $y(s+1)$  when updating the state estimate  $\hat{\boldsymbol{x}}(s+1)$ .

Define the state error as  $\boldsymbol{e}(s) := \boldsymbol{x}(s) - \hat{\boldsymbol{x}}(s) \in \mathbb{R}^n$ , and the posteriori covariance matrix as

$$\begin{aligned}\boldsymbol{P}(s) &:= \text{E}\{[\boldsymbol{x}(s) - \hat{\boldsymbol{x}}(s)][\boldsymbol{x}(s) - \hat{\boldsymbol{x}}(s)]^T\} \\ &= \text{E}\{\boldsymbol{e}(s)\boldsymbol{e}^T(s)\} \in \mathbb{R}^{n \times n}.\end{aligned}\tag{16}$$

Then the state estimation error covariance matrix  $\boldsymbol{P}(s)$  can be calculated by

$$\begin{aligned}\boldsymbol{P}(s+1) &= \text{E}\{[\boldsymbol{x}(s+1) - \hat{\boldsymbol{x}}(s+1)][\boldsymbol{x}(s+1) - \hat{\boldsymbol{x}}(s+1)]^T\} \\ &= \text{E}\{[\boldsymbol{A}\boldsymbol{e}(s) + \boldsymbol{B}_1\boldsymbol{e}(s-1) + \boldsymbol{B}_2\boldsymbol{e}(s-2) + \cdots \\ &\quad + \boldsymbol{B}_r\boldsymbol{e}(s-r) + \boldsymbol{w}(s) - \boldsymbol{K}(s+1)\boldsymbol{c}[\boldsymbol{x}(s+1) \\ &\quad - \hat{\boldsymbol{x}}_o(s+1)] - \boldsymbol{K}(s+1)v(s+1)][\boldsymbol{A}\boldsymbol{e}(s) + \boldsymbol{B}_1\boldsymbol{e}(s-1) \\ &\quad + \boldsymbol{B}_2\boldsymbol{e}(s-2) + \cdots + \boldsymbol{B}_r\boldsymbol{e}(s-r) + \boldsymbol{w}(s) - \boldsymbol{K}(s+1)\boldsymbol{c} \\ &\quad \times [\boldsymbol{x}(s+1) - \hat{\boldsymbol{x}}_o(s+1)] - \boldsymbol{K}(s+1)v(s+1)]^T\} \\ &= \text{E}\{[\{\boldsymbol{I} - \boldsymbol{K}(s+1)\boldsymbol{c}\}[\boldsymbol{x}(s+1) - \hat{\boldsymbol{x}}_o(s+1)] \\ &\quad - \boldsymbol{K}(s+1)v(s+1)]\{\boldsymbol{I} - \boldsymbol{K}(s+1)\boldsymbol{c}\}[\boldsymbol{x}(s+1) \\ &\quad - \hat{\boldsymbol{x}}_o(s+1)] - \boldsymbol{K}(s+1)v(s+1)]^T\} \\ &= [\boldsymbol{I} - \boldsymbol{K}(s+1)\boldsymbol{c}]\boldsymbol{P}_o(s+1)[\boldsymbol{I} - \boldsymbol{K}(s+1)\boldsymbol{c}]^T \\ &\quad + \boldsymbol{K}(s+1)\boldsymbol{R}_v\boldsymbol{K}^T(s+1).\end{aligned}\tag{17}$$

Because the system state  $\boldsymbol{x}(s)$ , and the state estimates  $\hat{\boldsymbol{x}}(s)$  and  $\hat{\boldsymbol{x}}_o(s)$  are uncorrected with  $\boldsymbol{w}(s)$ . That is to say,  $\text{E}\{\boldsymbol{x}(s)\boldsymbol{w}^T(s)\} = \mathbf{0}$ ,  $\text{E}\{\hat{\boldsymbol{x}}(s)\boldsymbol{w}^T(s)\} = \mathbf{0}$ ,  $\text{E}\{\boldsymbol{e}(s-i)\boldsymbol{w}^T(s)\} = \mathbf{0}$  ( $i = 0, 1, 2, \dots, r$ ). From (14), we have

$$\begin{aligned}\boldsymbol{P}_o(s+1) &= \boldsymbol{A}\text{E}\{\boldsymbol{e}(s)\boldsymbol{e}^T(s)\}\boldsymbol{A}^T + \boldsymbol{B}_1\text{E}\{\boldsymbol{e}(s-1)\boldsymbol{e}^T(s-1)\}\boldsymbol{B}_1^T + \cdots \\ &\quad + \boldsymbol{B}_r\text{E}\{\boldsymbol{e}(s-r)\boldsymbol{e}^T(s-r)\}\boldsymbol{B}_r^T + \boldsymbol{A}\text{E}\{\boldsymbol{e}(s) \\ &\quad \times \boldsymbol{e}^T(s-1)\}\boldsymbol{B}_1^T + \cdots + \boldsymbol{A}\text{E}\{\boldsymbol{e}(s)\boldsymbol{e}^T(s-r)\}\boldsymbol{B}_r^T \\ &\quad + \boldsymbol{B}_1\text{E}\{\boldsymbol{e}(s-1)\boldsymbol{e}^T(s)\}\boldsymbol{A}^T + \cdots + \boldsymbol{B}_1\text{E}\{\boldsymbol{e}(s-1) \\ &\quad \times \boldsymbol{e}^T(s-r)\}\boldsymbol{B}_r^T + \boldsymbol{B}_r\text{E}\{\boldsymbol{e}(s-r)\boldsymbol{e}^T(s)\}\boldsymbol{A}^T + \cdots \\ &\quad + \boldsymbol{B}_r\text{E}\{\boldsymbol{e}(s-r)\boldsymbol{e}^T(s-r+1)\}\boldsymbol{B}_{r-1}^T + \text{E}\{\boldsymbol{w}(s)\boldsymbol{w}^T(s)\}.\end{aligned}\tag{18}$$

Define the covariance matrix  $\boldsymbol{P}_{ij}(s) := \text{E}\{\boldsymbol{e}(s-i)\boldsymbol{e}(s-j)\}$ ,  $i = 0, 1, 2, \dots, r$ ,  $j = 0, 1, \dots, r$ . Then  $\boldsymbol{P}_o(s+1)$  can be expressed as

$$\boldsymbol{P}_o(s+1) = [\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2, \dots, \boldsymbol{B}_r]\boldsymbol{P}_1(s)[\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2, \dots, \boldsymbol{B}_r]^T + \boldsymbol{R}_w,\tag{19}$$

where  $\boldsymbol{P}_1(s) := [\boldsymbol{P}_{ij}(s)] \in \mathbb{R}^{(nr+n) \times (nr+n)}$ . In order to simplify the computation complexity, we assume that  $\boldsymbol{P}_{ij}(s) = 0$  ( $i \neq j$ ). Suppose that  $\boldsymbol{K}(s+1)$  is the optimal gain vector which minimizes the state estimation error covariance matrix  $\boldsymbol{P}(s+1)$ . In this condition, let  $\boldsymbol{P}(s+1)$  be the minimum covariance matrix. Obviously, if there exists the departure  $\delta\boldsymbol{K}(s+1)$  from the filtering gain to the optimal gain  $\boldsymbol{K}(s+1)$ , the state estimation error covariance matrix  $\boldsymbol{P}(s+1)$  computed by (17) will deviate from the minimum  $\boldsymbol{P}(s+1)$  and reach  $\boldsymbol{P}(s+1) + \delta\boldsymbol{P}(s+1)$ .  $\delta\boldsymbol{P}(s+1)$  is a

non-negative definite matrix. From (17), we find

$$\begin{aligned} & \mathbf{P}(s+1) + \delta\mathbf{P}(s+1) \\ &= \{\mathbf{I} - [\mathbf{K}(s+1) + \delta\mathbf{K}(s+1)]\mathbf{c}\}\mathbf{P}_o(s+1)\{\mathbf{I} - [\mathbf{K}(s+1) + \delta\mathbf{K}(s+1)]\mathbf{c}\}^T \\ & \quad + [\mathbf{K}(s+1) + \delta\mathbf{K}(s+1)]R_v[\mathbf{K}(s+1) + \delta\mathbf{K}(s+1)]^T, \end{aligned} \quad (20)$$

where  $\mathbf{P}(s+1)$  and  $\mathbf{K}(s+1)$  satisfy (17). By substituting (17) into (20), we obtain

$$\begin{aligned} \delta\mathbf{P}(s+1) &= \mathbf{M}(s+1) + \mathbf{M}^T(s+1) + \delta\mathbf{K}(s+1) \\ & \quad \times [\mathbf{cP}_o(s+1)\mathbf{c}^T + R_v]\delta\mathbf{K}^T(s+1), \end{aligned} \quad (21)$$

where

$$\begin{aligned} \mathbf{M}(s+1) &= -\delta\mathbf{K}(s+1)\{\mathbf{cP}_o(s+1)[\mathbf{I} - \mathbf{c}^T\mathbf{K}^T(s+1)] - R_v\mathbf{K}^T(s+1)\} \\ &= -\delta\mathbf{K}(s+1)\{\mathbf{cP}_o(s+1) - [\mathbf{cP}_o(s+1)\mathbf{c}^T + R_v]\mathbf{K}^T(s+1)\}. \end{aligned} \quad (22)$$

If we take

$$\mathbf{cP}_o(s+1) - [\mathbf{cP}_o(s+1)\mathbf{c}^T + R_v]\mathbf{K}^T(s+1) = 0,$$

then we can obtain

$$\mathbf{K}(s+1) = \mathbf{P}_o(s+1)\mathbf{c}^T[\mathbf{cP}_o(s+1)\mathbf{c}^T + R_v]^{-1}. \quad (23)$$

Thus, we have  $\mathbf{M}(s+1) = \mathbf{0}$  and

$$\delta\mathbf{P}(s+1) = \delta\mathbf{K}(s+1)[\mathbf{cP}_o(s+1)\mathbf{c}^T + R_v]\delta\mathbf{K}^T(s+1). \quad (24)$$

**Remark 4:** From (24), it is obvious that  $\mathbf{cP}_o(s+1)\mathbf{c}^T + R_v$  is non-negative. If  $\delta\mathbf{K}(s+1) \neq \mathbf{0}$ , then  $\delta\mathbf{P}(s+1) \neq \mathbf{0}$ , which shows that the non-negative deviation of  $\mathbf{P}(s+1)$  is generated when any departure  $\delta\mathbf{K}(s+1) \neq \mathbf{0}$ . Thus  $\mathbf{K}(s+1)$  in (23) is the optimal gain which makes  $\mathbf{P}(s+1)$  minimum.

Then the direct state estimation algorithm for the linear system with time-delays in (1)–(2) is as follows,

$$\hat{\mathbf{x}}_o(s+1) = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}_1\hat{\mathbf{x}}(s-1) + \cdots + \mathbf{B}_r\hat{\mathbf{x}}(s-r) + \mathbf{b}u(s), \quad (25)$$

$$\begin{aligned} \mathbf{P}_o(s+1) &= \mathbf{A}\mathbf{P}(s)\mathbf{A}^T + \mathbf{B}_1\mathbf{P}(s-1)\mathbf{B}_1^T + \mathbf{B}_2\mathbf{P}(s-2)\mathbf{B}_2^T + \cdots \\ & \quad + \mathbf{B}_r\mathbf{P}(s-r)\mathbf{B}_r^T + \mathbf{R}_w, \end{aligned} \quad (26)$$

$$\begin{aligned} \hat{\mathbf{x}}(s+1) &= \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}_1\hat{\mathbf{x}}(s-1) + \mathbf{B}_2\hat{\mathbf{x}}(s-2) + \cdots + \mathbf{B}_r\hat{\mathbf{x}}(s-r) + \mathbf{b}u(s) \\ & \quad + \mathbf{K}(s+1)[y(s+1) - \mathbf{c}\hat{\mathbf{x}}_o(s+1)], \end{aligned} \quad (27)$$

$$\mathbf{K}(s+1) = \mathbf{P}_o(s+1)\mathbf{c}^T[\mathbf{cP}_o(s+1)\mathbf{c}^T + R_v]^{-1}, \quad (28)$$

$$\begin{aligned} \mathbf{P}(s+1) &= [\mathbf{I} - \mathbf{K}(s+1)\mathbf{c}]\mathbf{P}_o(s+1)[\mathbf{I} - \mathbf{K}(s+1) \\ & \quad \times \mathbf{c}]^T + \mathbf{K}(s+1)R_v\mathbf{K}^T(s+1). \end{aligned} \quad (29)$$

**Remark 5:** In practical area, the variances of  $w(s)$  and  $v(s)$  are unknown. Thus the unknown  $\mathbf{R}_w$  and  $R_v$  in (25)–(29) may be replaced with their estimates  $\hat{\mathbf{R}}_w(s)$  and  $\hat{R}_v(s)$ :

$$\begin{aligned} \hat{\mathbf{R}}_w(s) &= \frac{1}{s} \sum_{j=1}^s [\hat{\mathbf{x}}(j+1) - \mathbf{A}\hat{\mathbf{x}}(j) - \sum_{i=1}^r \mathbf{B}_i\hat{\mathbf{x}}(j-i) - \mathbf{b}u(j)] \\ & \quad \times [\hat{\mathbf{x}}(j+1) - \mathbf{A}\hat{\mathbf{x}}(j) - \sum_{i=1}^r \mathbf{B}_i\hat{\mathbf{x}}(j-i) - \mathbf{b}u(j)]^T \in \mathbb{R}^{n \times n}, \end{aligned} \quad (30)$$

$$\hat{R}_v(s) = \frac{1}{s} \sum_{j=1}^s [y(j) - \mathbf{c}\hat{\mathbf{x}}(j)][y(j) - \mathbf{c}\hat{\mathbf{x}}(j)]^T \in \mathbb{R}. \quad (31)$$

Replacing  $\mathbf{R}_w$  and  $R_v$  in (25)–(29) with their estimates  $\hat{\mathbf{R}}_w(s)$  and  $\hat{R}_v(s)$  gives the direct state estimation algorithm:

$$\hat{\mathbf{x}}_o(s+1) = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}_1\hat{\mathbf{x}}(s-1) + \mathbf{B}_2\hat{\mathbf{x}}(s-2) + \cdots + \mathbf{B}_r\hat{\mathbf{x}}(s-r) + \mathbf{b}u(s), \quad (32)$$

$$\begin{aligned} \mathbf{P}_o(s+1) &= \mathbf{A}\mathbf{P}(s)\mathbf{A}^\top + \mathbf{B}_1\mathbf{P}(s-1)\mathbf{B}_1^\top + \mathbf{B}_2\mathbf{P}(s-2)\mathbf{B}_2^\top + \cdots \\ &\quad + \mathbf{B}_r\mathbf{P}(s-r)\mathbf{B}_r^\top + \hat{\mathbf{R}}_w(s), \end{aligned} \quad (33)$$

$$\begin{aligned} \hat{\mathbf{x}}(s+1) &= \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}_1\hat{\mathbf{x}}(s-1) + \mathbf{B}_2\hat{\mathbf{x}}(s-2) + \cdots + \mathbf{B}_r\hat{\mathbf{x}}(s-r) + \mathbf{b}u(s) \\ &\quad + \mathbf{K}(s+1)[y(s+1) - \mathbf{c}\hat{\mathbf{x}}_o(s+1)], \end{aligned} \quad (34)$$

$$\mathbf{K}(s+1) = \mathbf{P}_o(s+1)\mathbf{c}^\top[\mathbf{c}\mathbf{P}_o(s+1)\mathbf{c}^\top + \hat{R}_v(s)]^{-1}, \quad (35)$$

$$\begin{aligned} \mathbf{P}(s+1) &= [\mathbf{I} - \mathbf{K}(s+1)\mathbf{c}]\mathbf{P}_o(s+1)[\mathbf{I} - \mathbf{K}(s+1)\mathbf{c}]^\top \\ &\quad + \mathbf{K}(s+1)\hat{R}_v(s)\mathbf{K}^\top(s+1). \end{aligned} \quad (36)$$

The estimation steps of the algorithm in (30)–(36) to compute the state estimate  $\hat{\mathbf{x}}(s)$  of  $\mathbf{x}(s)$  are listed in the following.

1. Let  $s = 1$ , set the initial values  $\hat{\mathbf{x}}(1) = \mathbf{1}_n$ ,  $\mathbf{P}(1) = \mathbf{I}_n$ ,  $u(s-i) = 0$ ,  $y(s-i) = 0$  and  $\hat{\mathbf{x}}(s-i) = \mathbf{0}$  for  $i = 1, 2, \dots, n$ ,  $\hat{R}_v(s) = 1$  and  $\hat{\mathbf{R}}_w(s) = \mathbf{I}_n$ .
2. Collect the input-output information  $u(s)$  and  $y(s)$  and obtain the system parameters  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{B}_j$  for  $j = 1, 2, \dots, r$ ,  $\mathbf{b}$ , and  $\mathbf{c}$ .
3. Compute the priori state estimate  $\hat{\mathbf{x}}_o(s+1)$  by (32).
4. Compute the priori state estimation error covariance matrix  $\mathbf{P}_o(s+1)$  using (33).
5. Compute the gain vector  $\mathbf{K}(s+1)$  by (35) and the posteriori state estimation error covariance matrix  $\mathbf{P}(s+1)$  by (36). Increase  $s$  by 1.
6. Collect the input-output data  $u(s)$  and  $y(s)$  and update the state estimation vector  $\hat{\mathbf{x}}(s+1)$  using (34).
7. Compute the covariance matrix  $\hat{\mathbf{R}}_w(s)$  by (30) and the variance  $\hat{R}_v(s)$  by (31).
8. Go to Step 2 and continue the recursive calculation.

**Remark 6:** The computational burden may be evaluated by floating point operation [47]. Tables I and II give the number of additions and multiplications for each recursive computation of the generalized Kalman filtering algorithm and the director state estimation algorithm for computational complex analysis.

Table I. The computational efficiency of the generalized Kalman filtering algorithm

Variables	Computation Sequences	Number of multiplications	Number of additions
$\hat{\mathbf{X}}(s+1 s)$	$\hat{\mathbf{X}}(s+1 s) = \mathbf{G}\hat{\mathbf{X}}(s s) + \mathbf{b}u(s) \in \mathbb{R}^{nr+n}$	$(nr+n)^2 + nr+n$	$(nr+n)^2$
$\mathbf{P}(s+1 s)$	$\mathbf{P}(s+1 s) = \mathbf{W}(s)\mathbf{G}^\top + \hat{\mathbf{R}}_w(s) \in \mathbb{R}^{(nr+n) \times (nr+n)}$ $\mathbf{W}(s) := \mathbf{G}\mathbf{P}(s) \in \mathbb{R}^{(nr+n) \times (nr+n)}$	$(nr+n)^3$ $(nr+n)^3$	$(nr+n)^3$ $(nr+n)^3 - (nr+n)^2$
$\hat{\mathbf{X}}(s+1 s+1)$	$\hat{\mathbf{X}}(s+1 s+1) = \hat{\mathbf{X}}(s+1 s) + \mathbf{K}(s+1)s(s+1) \in \mathbb{R}^{nr+n}$ $s(s+1) := y(s+1) - \mathbf{F}\hat{\mathbf{X}}(s+1 s) \in \mathbb{R}$	$nr+n$ $nr+n$	$nr+n$ $nr+n$
$\mathbf{K}(s+1)$	$\mathbf{K}(s+1) = \mathbf{L}(s+1)/\zeta(s+1) \in \mathbb{R}^{nr+n}$ $\zeta(s+1) := \mathbf{F}\mathbf{L}(s+1) + \hat{R}_v(s) \in \mathbb{R}$ $\mathbf{L}(s+1) := \mathbf{P}(s+1 s)\mathbf{F}^\top \in \mathbb{R}^{nr+n}$	$nr+n$ $nr+n$ $(nr+n)^2$	0 $nr+n$ $(nr+n)^2 - (nr+n)$
$\mathbf{P}(s+1 s+1)$	$\mathbf{P}(s+1 s+1) = [\mathbf{I} - \mathbf{K}(s+1)\mathbf{F}]\mathbf{P}(s s)[\mathbf{I} - \mathbf{K}(s+1)\mathbf{F}]^\top$ $+ \mathbf{K}(s+1)\hat{R}_v(s)\mathbf{K}^\top(s+1) \in \mathbb{R}^{(nr+n) \times (nr+n)}$	$(nr+n)^3 + 4(nr+n)^2$ $+(nr+n)$	$(nr+n)^3 + 3(nr+n)^2$
$\hat{\mathbf{R}}_w(s)$	$\hat{\mathbf{R}}_w(s) = \frac{1}{s} \sum_{j=1}^s [\mathbf{X}'(j)\mathbf{X}'^\top(j)] \in \mathbb{R}^{(nr+n) \times (nr+n)}$ $\mathbf{X}'(s) := \hat{\mathbf{X}}(s+1 s+1) - \mathbf{G}\hat{\mathbf{X}}(s s) - \mathbf{H}u(s) \in \mathbb{R}^{nr+n}$	$2(nr+n)^2$ $(nr+n)^2 + (nr+n)$	$(nr+n)^2$ $(nr+n)^2 + (nr+n)$
$\hat{R}_v(s)$	$\hat{R}_v(s) = \frac{1}{s} \sum_{j=1}^s \varrho^2(j) \in \mathbb{R}$ $\varrho(s) := y(s) - \mathbf{F}\hat{\mathbf{X}}(s s) \in \mathbb{R}$	2 $nr+n$	1 $nr+n$
Sum		$3(nr+n)^3 + 9(nr+n)^2$ $+ 8(nr+n) + 2$	$3(nr+n)^3 + 3(nr+n)^2$ $+ 4(nr+n) + 1$
Total flop		$N_1 := 6(nr+n)^3 + 12(nr+n)^2 + 12(nr+n) + 3$	

The difference of the computation cost between two algorithms at each step is

$$N_1 - N_2 = 6(nr+n)^3 + 12(nr+n)^2 + 12(nr+n) + 3$$

Table II. The computational efficiency of the direct state estimation algorithm based on the delta operator

Variables	Computation Sequences	Number of multiplications	Number of additions
$\hat{\mathbf{x}}_o(s+1)$	$\hat{\mathbf{x}}_o(s+1) = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}_1\hat{\mathbf{x}}(s-1) + \dots + \mathbf{B}_r\hat{\mathbf{x}}(s-r) + \mathbf{b}u(s) \in \mathbb{R}^n$	$n^2(r+1) + n$	$n^2(r+1)$
$\mathbf{P}_o(s+1)$	$\mathbf{P}_o(s+1) = \mathbf{W}_1(s) + \dots + \mathbf{W}_{r+1}(s) + \hat{\mathbf{R}}_w(s) \in \mathbb{R}^{n \times n}$ $\mathbf{W}_1(s) := \mathbf{A}\mathbf{P}(s)\mathbf{A}^T \in \mathbb{R}^{n \times n}$ $\mathbf{W}_2(s) := \mathbf{B}_1\mathbf{P}(s-1)\mathbf{B}_1^T \in \mathbb{R}^{n \times n}$ $\vdots$ $\mathbf{W}_{r+1}(s) := \mathbf{B}_r\mathbf{P}(s-r)\mathbf{B}_r^T \in \mathbb{R}^{n \times n}$	0 $2n^3$ $2n^3$ $\vdots$ $2n^3$	$(r+1)n^2$ $2n^3 - 2n^2$ $2n^3 - 2n^2$ $\vdots$ $2n^3 - 2n^2$
$\hat{\mathbf{x}}(s+1)$	$\hat{\mathbf{x}}(s+1) = \hat{\mathbf{x}}_o(s+1) + \mathbf{K}(s+1)s(s+1) \in \mathbb{R}^n$ $s(s+1) := y(s+1) - \mathbf{c}\hat{\mathbf{x}}_o(s+1) \in \mathbb{R}$	$n$ $n$	$n$ $n$
$\mathbf{K}(s+1)$	$\mathbf{K}(s+1) = \mathbf{L}(s+1)/\zeta(s+1) \in \mathbb{R}^n$ $\zeta(s+1) := \mathbf{c}\mathbf{L}(s+1) + \hat{\mathbf{R}}_v(s) \in \mathbb{R}$ $\mathbf{L}(s+1) := \mathbf{P}_o(s+1)\mathbf{c}^T \in \mathbb{R}^n$	$n$ $n$ $n^2$	0 $n$ $n^2 - n$
$\mathbf{P}(s+1)$	$\mathbf{P}(s+1) = [\mathbf{I} - \mathbf{K}(s)\mathbf{c}]\mathbf{P}_o(s)[\mathbf{I} - \mathbf{K}(s)\mathbf{c}]^T + \mathbf{K}(s)\hat{\mathbf{R}}_v(s)\mathbf{K}^T(s) \in \mathbb{R}^{n \times n}$	$n^3 + 4n^2 + n$	$n^3 + 3n^2$
$\hat{\mathbf{R}}_w(s)$	$\hat{\mathbf{R}}_w(s) = \frac{1}{s} \sum_{j=1}^s [\mathbf{x}'(j)\mathbf{x}'^T(j)] \in \mathbb{R}^{n \times n}$ $\mathbf{x}'(s) := \hat{\mathbf{x}}(s+1) - \mathbf{A}\hat{\mathbf{x}}(s) - \mathbf{B}_1\hat{\mathbf{x}}(s-1) - \dots - \mathbf{B}_r\hat{\mathbf{x}}(s-r) - \mathbf{b}u(s) \in \mathbb{R}^n$	$2n^2$ $(r+1)n^2 + n$	$n^2$ $(r+1)n^2 + n$
$\hat{\mathbf{R}}_v(s)$	$\hat{\mathbf{R}}_v(s) = \frac{1}{s} \sum_{j=1}^s \varrho^2(j) \in \mathbb{R}$ $\varrho(s) := y(s) - \mathbf{c}\hat{\mathbf{x}}(s) \in \mathbb{R}$	2 $n$	1 $n$
Sum		$(2r+3)n^3 + (2r+9)n^2 + 8n + 2$	$(2r+3)n^3 + (r+6)n^2 + 4n + 1$
Total flop		$N_2 := (4r+6)n^3 + (3r+15)n^2 + 12n + 3$	

$$\begin{aligned}
& -[(4r+6)n^3 + (3r+15)n^2 + 12n + 3] \\
& = n^3(6r^3 + 18r^2 + 14r) + n^2(12r^2 \\
& \quad + 21r - 1) + 12nr.
\end{aligned}$$

Assume that the system order  $n = 10$  and the delay  $r = 5$ , then we can calculate the difference between the computation loads of the two algorithms at each step:

$$\begin{aligned}
N_1 - N_2 &= (6 \times 60^3 + 12 \times 60^2 + 723) \\
& \quad - (26 \times 10^3 + 30 \times 10^2 + 123) \\
&= 1.3108 \times 10^6.
\end{aligned}$$

Assume that the system order  $n = 100$  and the delay  $r = 50$ , then the difference between two algorithms is

$$\begin{aligned}
N_1 - N_2 &= (6 \times 5100^3 + 12 \times 5100^2 + 12 \times 5100 + 3) \\
& \quad - (206 \times 100^3 + 165 \times 100^2 + 12 \times 100 + 3) \\
&\approx 8.0 \times 10^{11}.
\end{aligned}$$

Table III. The computational comparison between two algorithms

$n$	$r$	$N_1 - N_2$	$(N_1 - N_2)/N_1$	$N_2/N_1$
10	10	$8.08 \times 10^6$	0.993	0.007
20	10	$6.41 \times 10^7$	0.994	0.006
20	20	$4.50 \times 10^8$	0.998	0.002

**Remark 7:** Tables I–III show the comparison of the computational burden of the two algorithms, which illustrates that the direct state estimation algorithm based on the delta operator can greatly reduce the computational burden compared with the generalized Kalman filtering algorithm, especially for large-scale systems.



4. NUMERICAL EXAMPLES

**Example 1.** Consider a third-order system with time-delay:

$$\begin{aligned} \mathbf{x}(s+1) &= \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(s) \\ \mathbf{x}_2(s) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{B}_{11} \\ \mathbf{B}_{12} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(s-1) \\ \mathbf{x}_2(s-1) \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} u(s) + \mathbf{w}(s), \\ y(s) &= \mathbf{c}\mathbf{x}(s) + v(s), \end{aligned}$$

where the parameters are given by

$$\begin{aligned} \mathbf{A}_{11} &= 0.10, \quad \mathbf{A}_{12} = [0.40, 0.20], \\ \mathbf{A}_{21} &= \begin{bmatrix} -0.30 \\ -0.20 \end{bmatrix}, \quad \mathbf{A}_{22} = \begin{bmatrix} 0.20 & -0.40 \\ 0.10 & -0.10 \end{bmatrix}, \\ \mathbf{B}_{11} &= [0.20, 0.60, 0.30], \\ \mathbf{B}_{12} &= \begin{bmatrix} 0.20 & -0.20 & -0.30 \\ -0.40 & -0.20 & -0.10 \end{bmatrix}, \\ \mathbf{b} &= [0.30, 0.50, 0.60]^T, \quad \mathbf{c} = [0.40, 0.60, 0.50]. \end{aligned}$$

In simulation, the input  $\{u(s)\}$  is taken as a persistent excitation signal sequence with zero mean and unit variance, and  $\{v(s)\}$  as a random sequence with the normal distribution, zero mean and variance  $R_v = 0.10^2$ , and  $\{\mathbf{w}(s)\}$  as a white noise vector sequence with zero mean and variance  $\mathbf{R}_w = 0.05^2 \mathbf{I}_3$ . Take the data length  $L = 300$  as the data length, and apply the direct state estimation algorithm in (30)–(36) to estimate the states of the considered system. The system input  $u(s)$ , the true output  $y(s)$  and the predicted output  $\hat{y}(s)$  are shown in Figure 1. The states  $\mathbf{x}(s)$  and their estimates  $\hat{\mathbf{x}}(s)$  and errors versus  $s$  by the generalized Kalman filtering algorithm are shown in Figures 2–4. The states  $\mathbf{x}(s)$  and their estimates  $\hat{\mathbf{x}}(s)$  and errors versus  $s$  by the direct state estimation algorithm are shown in Figures 5–7. The root mean squares error is used to describe the error between the true state  $x_i(s)$  and its estimated value  $\hat{x}_i(s)$ , and the error between the true output  $y(s)$  and its predicted output  $\hat{y}(s)$ , which are defined as

$$\begin{aligned} Error_x &= \left\{ \frac{1}{L} \sum_{s=1}^L [\hat{x}_i(s) - x_i(s)]^2 \right\}^{1/2}, \\ Error_y &= \left\{ \frac{1}{L} \sum_{s=1}^L [\hat{y}(s) - y(s)]^2 \right\}^{1/2}. \end{aligned}$$

The results are shown in Tables IV and V.

Table IV. The root mean squares error of the generalized Kalman filtering algorithm

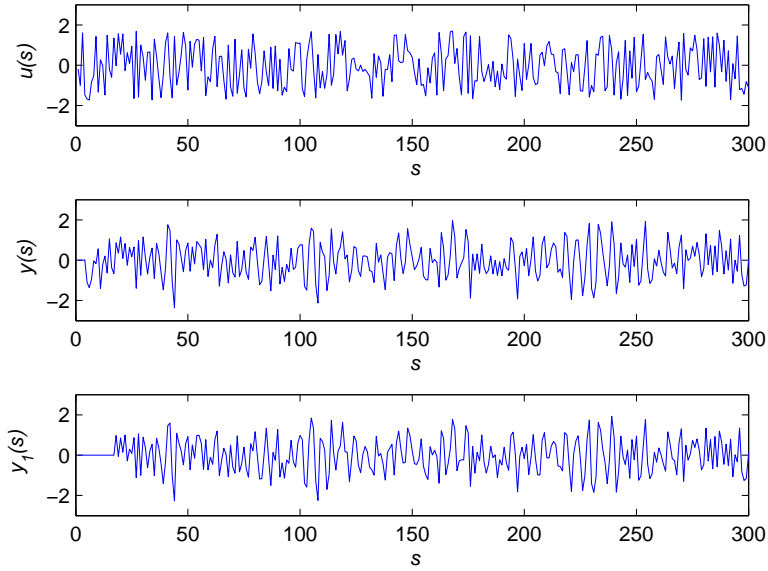
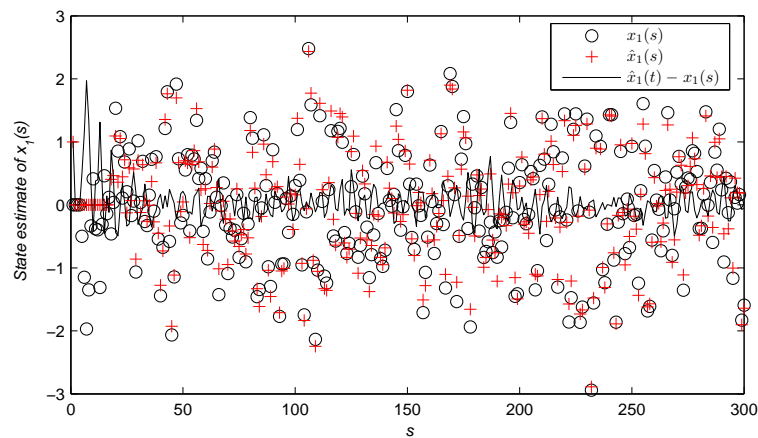
$R_v$	$\mathbf{R}_w$	$x_1(s)$	$x_2(s)$	$x_3(s)$	$y(s)$
$0.10^2$	$0.05^2 \mathbf{I}_6$	0.22360	0.21388	0.24850	0.21475
$0.05^2$	$0.10^2 \mathbf{I}_6$	0.28187	0.27252	0.27762	0.22835

**Example 2.** Consider the following system:

$$\begin{aligned} \mathbf{x}(s+1) &= \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(s) \\ \mathbf{x}_2(s) \\ \mathbf{x}_3(s) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{B}_{11} \\ \mathbf{B}_{12} \\ \mathbf{B}_{13} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(s-1) \\ \mathbf{x}_2(s-1) \\ \mathbf{x}_3(s-1) \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix} u(s) + \mathbf{w}(s), \end{aligned}$$

Table V. The root mean squares error of the direct state estimation algorithm

$R_v$	$R_w$	$x_1(s)$	$x_2(s)$	$x_3(s)$	$y(s)$
$0.10^2$	$0.05^2 I_3$	0.21313	0.19787	0.24351	0.17861
$0.15^2$	$0.05^2 I_3$	0.22125	0.20553	0.24756	0.19423
$0.20^2$	$0.05^2 I_3$	0.23125	0.21491	0.25275	0.21448
$0.05^2$	$0.10^2 I_3$	0.22617	0.20542	0.24981	0.16904
$0.05^2$	$0.15^2 I_3$	0.25340	0.22602	0.26550	0.17085
$0.05^2$	$0.20^2 I_3$	0.28617	0.25130	0.28650	0.17359

Figure 1. The system input  $u(s)$ , output  $y(s)$  and the predicted output  $\hat{y}(s)$  versus  $s$ Figure 2. State  $x_1(s)$  and the estimated state  $\hat{x}_1(s)$  versus  $s$  by the generalized Kalman filtering algorithm

$$y(s) = \mathbf{c}\mathbf{x}(s) + v(s),$$

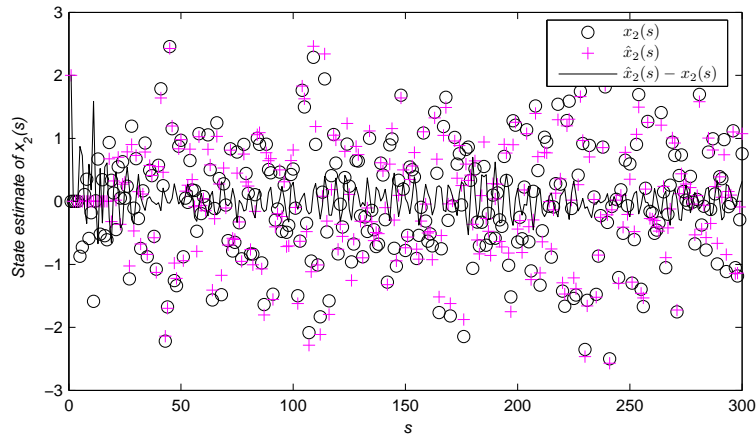


Figure 3. State  $x_2(s)$  and the estimated state  $\hat{x}_2(s)$  versus  $s$  by the generalized Kalman filtering algorithm

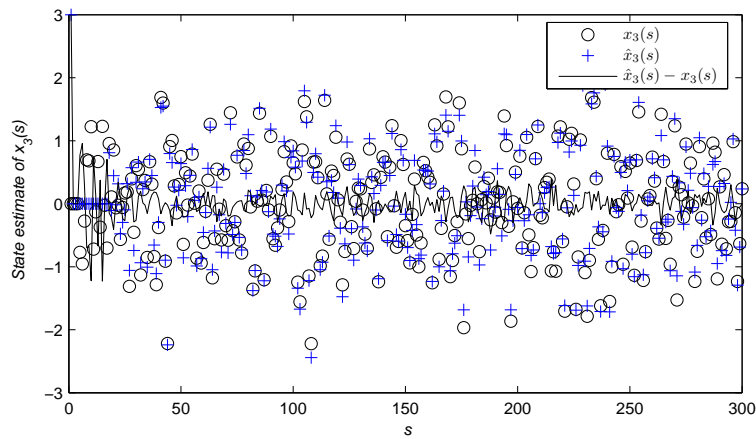


Figure 4. State  $x_3(s)$  and the estimated state  $\hat{x}_3(s)$  versus  $s$  by the generalized Kalman filtering algorithm

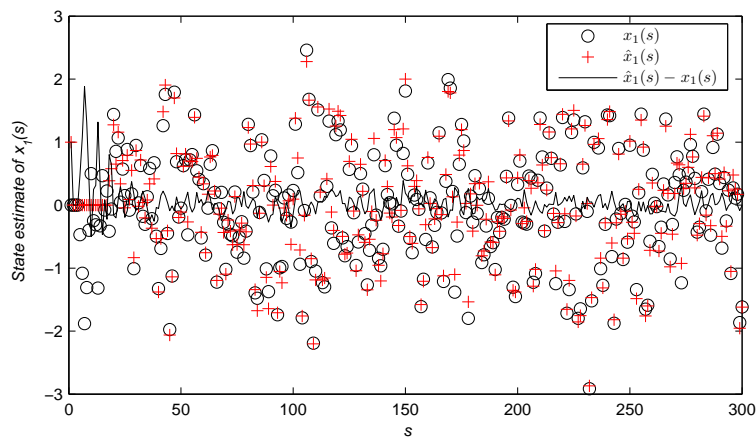


Figure 5. State  $x_1(s)$  and the estimated state  $\hat{x}_1(s)$  versus  $s$  by the direct state estimation algorithm

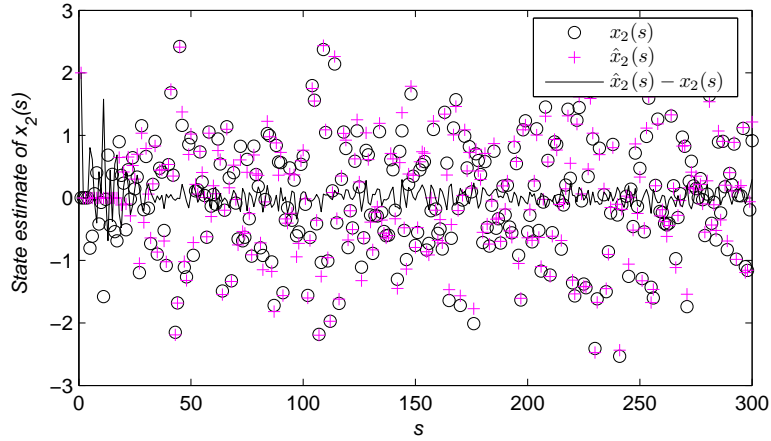


Figure 6. State  $x_2(s)$  and the estimated state  $\hat{x}_2(s)$  versus  $s$  by the direct state estimation algorithm

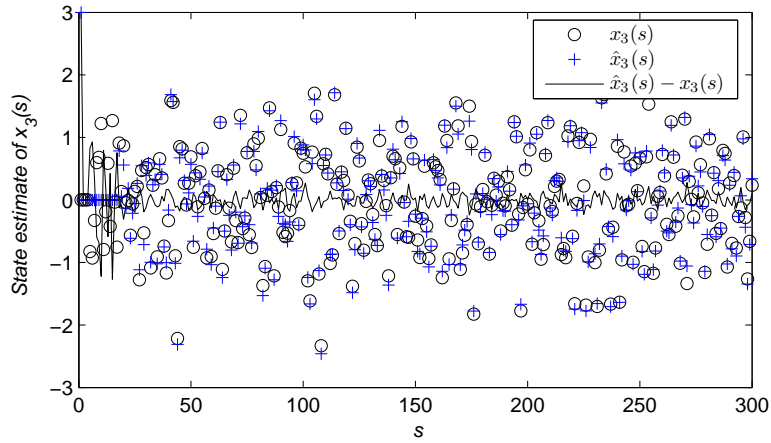


Figure 7. State  $x_3(s)$  and the estimated state  $\hat{x}_3(s)$  versus  $s$  by the direct state estimation algorithm

where the parameters are given by

$$\begin{aligned}
 \mathbf{A}_{11} &= 0.10, \quad \mathbf{A}_{12} = [0.20, -0.20], \quad \mathbf{A}_{13} = [0.50, -0.30, 0.20], \quad \mathbf{A}_{21} = \begin{bmatrix} -0.30 \\ 0.20 \end{bmatrix}, \\
 \mathbf{A}_{22} &= \begin{bmatrix} 0.20 & -0.22 \\ 0.30 & -0.10 \end{bmatrix}, \quad \mathbf{A}_{23} = \begin{bmatrix} 0.25 & 0.15 & -0.25 \\ 0.30 & -0.15 & 0.40 \end{bmatrix}, \quad \mathbf{A}_{31} = \begin{bmatrix} 0.14 \\ -0.15 \\ -0.25 \end{bmatrix}, \\
 \mathbf{A}_{32} &= \begin{bmatrix} -0.12 & 0.25 \\ 0 & 0.22 \\ -0.20 & 0.30 \end{bmatrix}, \quad \mathbf{A}_{33} = \begin{bmatrix} 0 & 0.30 & -0.21 \\ 0.25 & 0.20 & -0.30 \\ -0.20 & 0.35 & 0.15 \end{bmatrix}, \\
 \mathbf{B}_{11} &= [0.20, 0.40, 0, 0.10, 0.30, -0.25], \\
 \mathbf{B}_{12} &= \begin{bmatrix} 0.32 & 0.12 & -0.25 & 0.25 & 0.30 & 0.20 \\ -0.35 & 0.15 & -0.15 & 0.25 & 0 & 0.25 \end{bmatrix}, \\
 \mathbf{B}_{13} &= \begin{bmatrix} -0.14 & -0.22 & 0.15 & 0.26 & 0.43 & -0.15 \\ 0.20 & 0.30 & 0 & -0.25 & 0.10 & 0.25 \\ -0.20 & -0.30 & -0.45 & 0.15 & 0.45 & 0.25 \end{bmatrix},
 \end{aligned}$$

$$\mathbf{b}_1 = 0.3, \quad \mathbf{b}_2 = [0.5, 0.6]^T, \quad \mathbf{b}_3 = [0.35, 0.40, 0.60]^T, \\ \mathbf{c} = [0.4, 0.6, 0.5, 0.3, 0.2, 0.1].$$

The simulation condition is the same as that in Example 1. Take  $\{v(s)\}$  as a random sequence with the normal distribution, zero mean and variance  $R_v = 0.10^2$ , and  $\{w(s)\}$  as a white noise vector sequence with zero mean and variance  $R_w = 0.05^2 \mathbf{I}_6$ . Take the data length  $L = 300$  as the data length, and apply the direct state estimation algorithm in (30)–(36) to estimate the states of the considered system. The system input  $u(s)$ , the true output  $y(s)$  and the predicted output  $\hat{y}(s)$  are shown in Figure 8. The state  $x(s)$  and their estimates  $\hat{x}(s)$  and errors versus  $s$  are shown in Figures 9–11 and Table VI.

Table VI. The root mean squares error of the algorithm

$R_v$	$R_w$	$x_1(s)$	$x_2(s)$	$x_3(s)$	$x_4(s)$	$x_5(s)$	$x_6(s)$	$y(s)$
$0.10^2$	$0.05^2 \mathbf{I}_6$	0.12170	0.16359	0.20876	0.10843	0.14794	0.20944	0.15523
$0.15^2$	$0.05^2 \mathbf{I}_6$	0.13191	0.17089	0.21133	0.11074	0.15249	0.21055	0.16663
$0.20^2$	$0.05^2 \mathbf{I}_6$	0.14383	0.17980	0.21433	0.11396	0.15810	0.21204	0.18057
$0.05^2$	$0.10^2 \mathbf{I}_6$	0.16699	0.17819	0.23226	0.15911	0.17552	0.25575	0.14069
$0.05^2$	$0.15^2 \mathbf{I}_6$	0.23150	0.20952	0.27084	0.22020	0.21885	0.31968	0.13729
$0.05^2$	$0.20^2 \mathbf{I}_6$	0.29976	0.24777	0.31760	0.28459	0.26852	0.39236	0.13664

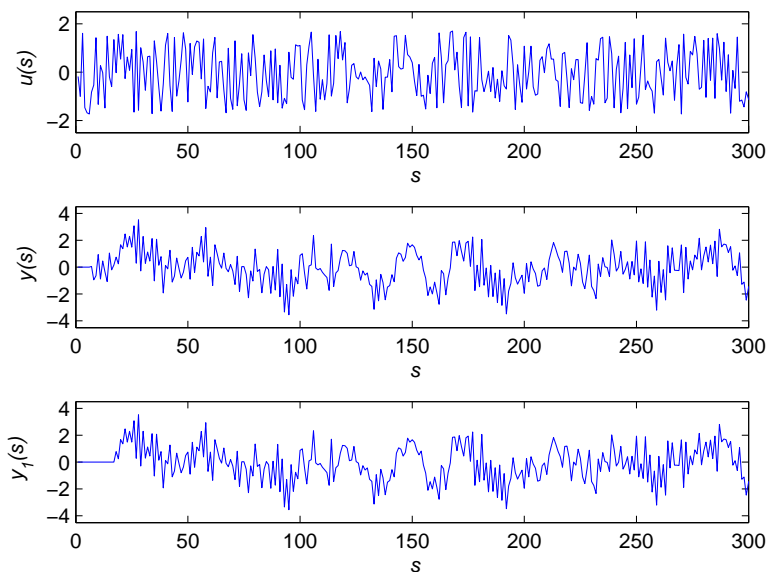


Figure 8. The system input  $u(s)$ , output  $y(s)$  and the predicted output  $\hat{y}(s)$  versus  $s$

From the simulation results in Tables I–V and Figures 1–11, we can draw the following conclusions.

- The state estimation accuracy of the direct state estimation algorithm is approximate to that of the generalized Kalman filtering algorithm. They both has good performance because the estimated states are close to their true values with  $s$  increasing. However, the direct state estimation algorithm requires less computational cost compared with the generalized Kalman filtering algorithm, see Tables I–V, and Figures 2–7 and Figures 9–11.

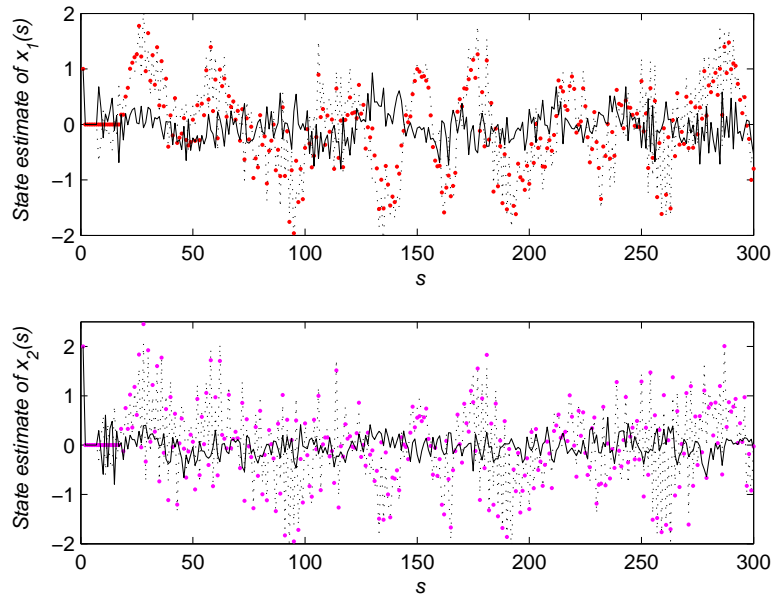


Figure 9. States  $x_1(s)$ ,  $x_2(s)$  and the estimated states  $\hat{x}_1(s)$  and  $\hat{x}_2(s)$  versus  $s$

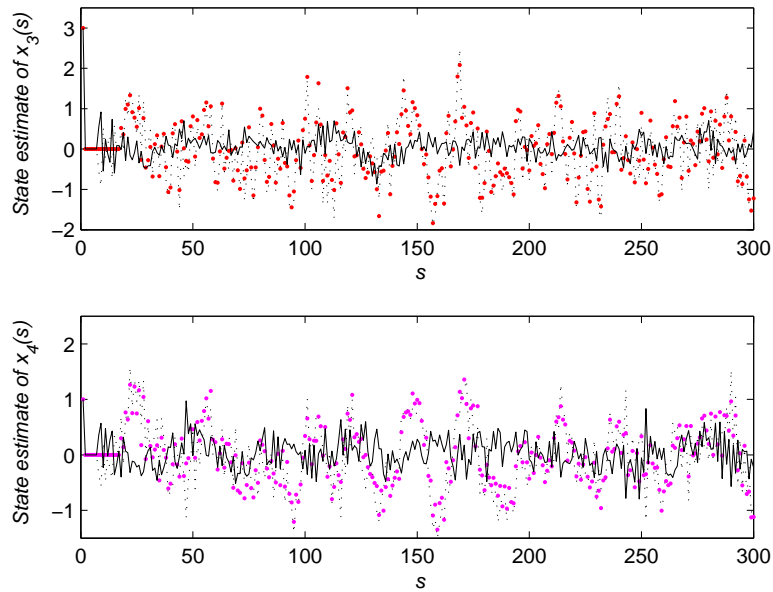


Figure 10. States  $x_3(s)$ ,  $x_4(s)$  and the estimated states  $\hat{x}_3(s)$  and  $\hat{x}_4(s)$  versus  $s$

- The proposed estimation algorithm can generate good estimates because the predicted output is close to the true output, see Tables V–VI, and Figures 1 and 8.
- The state estimation accuracy of the proposed algorithm becomes better under the lower noise levels, see Tables V–VI.

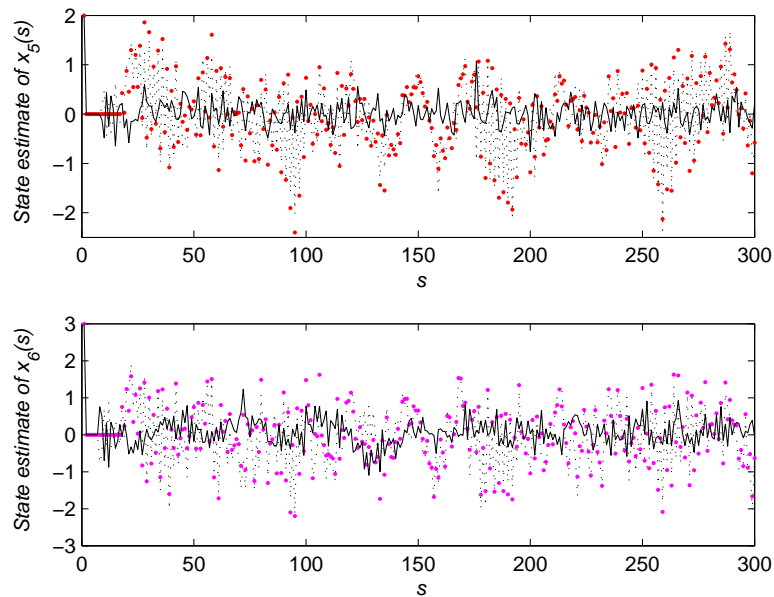


Figure 11. States  $x_5(s)$ ,  $x_6(s)$  and the estimated states  $\hat{x}_5(s)$  and  $\hat{x}_6(s)$  versus  $s$

## 5. CONCLUSIONS

This paper studies the state estimation for linear systems with multi-state-delays from observation data. A state estimation algorithm based on the Kalman filtering principle is proposed for comparison. A direct state estimation algorithm is presented by minimizing the state estimation error covariance matrix based on the delta operator. The computational complexity analysis shows that the direct state estimation algorithm requires less computational cost than the generalized state estimation algorithm. The simulation results show that the direct state estimation algorithm can generate accurate estimates. The methods proposed in this paper can combine some statistical methods [48-55] to study the parameter identification and state filter design for different systems with coloured noise [56-62].

## ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under grant number 61873111, and the 111 Project (B12018) and the Postgraduate Research and Practice Innovation Program of Jiangsu Province under grant number KYCX18-1854.

## REFERENCES

1. Svensson A, Schön TB. A flexible state-space model for learning nonlinear dynamical systems. *Automatica*. 2017;80:189-199.
2. Xu L, Ding F, Zhu, QM. Hierarchical Newton and least squares iterative estimation algorithm for dynamic systems by transfer functions based on the impulse responses. *International Journal of Systems Science*. 2019;50(1):141-151.
3. Xu L, Ding F. Parameter estimation algorithms for dynamical response signals based on the multi-innovation theory and the hierarchical principle. *IET Signal Processing*. 2017;11(2):228-237.
4. Xu L. Application of the Newton iteration algorithm to the parameter estimation for dynamical systems. *Journal of Computational and Applied Mathematics*. 2015;288:33-43.
5. Pan J, Ma H, Jiang X, et al. Adaptive gradient-based iterative algorithm for multivariate controlled autoregressive moving average systems using the data filtering technique. *Complexity*. 2018. Article ID 9598307. <https://doi.org/10.1155/2018/9598307>

6. Wan XK, Wu H, Qiao F, et al. Electrocardiogram baseline wander suppression based on the combination of morphological and wavelet transformation based filtering. *Computational and Mathematical Methods in Medicine*. 2019. Article ID 7196156. <https://doi.org/10.1155/2019/7196156>
7. Chen J, Liu Y, Xu L. A new filter-based stochastic gradient algorithm for dual-rate ARX models. *International Journal of Adaptive Control and Signal Processing*. 2018;32(11):1557-1568.
8. Wang YJ, Ding F. A filtering based multi-innovation gradient estimation algorithm and performance analysis for nonlinear dynamical systems. *IMA Journal of Applied Mathematics*. 2017;82(6):1171-1191.
9. Xu L, Ding F, Gu Y, Alsaedi A, Hayat T. A multi-innovation state and parameter estimation algorithm for a state space system with d-step state-delay. *Signal Processing*. 2017;140:97-103.
10. Schön T, Wills A, Ninness B. System identification of nonlinear state-space models. *Automatica*. 2011;47(1):39-49.
11. Partovibakhsh M, Liu G. An adaptive unscented Kalman filtering approach for online estimation of model parameters and state-of-charge of lithium-ion batteries for autonomous mobile robots. *IEEE Transactions on Control Systems Technology*. 2015;23(1):357-363.
12. Xu L, Xiong WL, A. Alsaedi A, Hayat T. Hierarchical parameter estimation for the frequency response based on the dynamical window data. *International Journal of Control Automation and Systems*. 2018;16(4):1756-1764.
13. Xu L, Ding F. Iterative parameter estimation for signal models based on measured data. *Circuits Systems and Signal Processing*. 2018;37(7):3046-3069.
14. Xu L, Ding F. Recursive least squares and multi-innovation stochastic gradient parameter estimation methods for signal modeling. *Circuits Systems and Signal Processing*. 2017;36(4):1735-1753.
15. Liu SY, Ding F, Xu L, Hayat T. Hierarchical principle-based iterative parameter estimation algorithm for dual-frequency signals. *Circuits Systems and Signal Processing*. 2019;38. <https://doi.org/10.1007/s00034-018-1015-1>
16. Xu L. The parameter estimation algorithms based on the dynamical response measurement data. *Advances in Mechanical Engineering*. 2017;9(11):1-12. doi: 10.1177/1687814017730003
17. Xu L. The damping iterative parameter identification method for dynamical systems based on the sine signal measurement. *Signal Processing*. 2016;120:660-667.
18. Xu L, Ding F. Parameter estimation for control systems based on impulse responses. *International Journal of Control Automation and Systems*. 2017;15(6):2471-2479.
19. Xu L, Chen L, Xiong WL. Parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration. *Nonlinear Dynamics*. 2015;79(3):2155-2163.
20. Chen J, Huang B, Ding F, Gu Y. Variational Bayesian approach for ARX systems with missing observations and varying time-delays. *Automatica*. 2018;94:194-204.
21. Gu Y, Chou Y, Liu J, Ji Y. Moving horizon estimation for multirate systems with time-varying time-delays. *Journal of the Franklin Institute*. 2019;356(4):2325-2345.
22. Pan J, Li W, Zhang HP. Control algorithms of magnetic suspension systems based on the improved double exponential reaching law of sliding mode control. *International Journal of Control Automation and Systems*. 2018;16(6):2878-2887.
23. Chen J, Zhu QM, Li J, Liu YJ. Biased compensation recursive least squares-based threshold algorithm for time-delay rational models via redundant rule. *Nonlinear Dynamics*. 2018;91(2):797-807.
24. Shi P, Zhang Y, Agarwal RK. Stochastic finite-time state estimation for discrete time-delay neural networks with Markovian jumps. *Neurocomputing*. 2015;151:168-174.
25. Xu L. A proportional differential control method for a time-delay system using the Taylor expansion approximation. *Applied Mathematics and Computation*. 2014;236:391-399.
26. Gu Y, Ding F, Li JH. States based iterative parameter estimation for a state space model with multi-state delays using decomposition. *Signal Processing*. 2015;106:294-300.
27. Chen GY, Gan M, Chen CP, Li HX. A regularized variable projection algorithm for separable nonlinear least-squares problems. *IEEE Transactions on Automatic Control*. 2019;64(2):526-537.
28. Chen GY, Gan M, Ding F, Chen CP. Modified Gram-Schmidt method-based variable projection algorithm for separable nonlinear models. *IEEE Transactions on Neural Networks and Learning Systems*. (2019). doi: 10.1109/TNNLS.2018.2884909
29. Gan M, Chen CPL, Chen GY, Chen L. On some separated algorithms for separable nonlinear least squares problems. *IEEE Transactions on Cybernetics*. 2018;48(10):2866-2874.
30. Gan M, Li HX. An efficient variable projection formulation for separable nonlinear least squares problems. *IEEE Transactions on Cybernetics*. 2014;44(5):707-711.
31. Gan M, Li HX, Peng H. A variable projection approach for efficient estimation of RBF-ARX model. *IEEE Transactions on Cybernetics*. 2015;45(3):462-471.
32. Gan M, Chen CLP, Li HX, Chen L. Gradient radial basis function based varying-coefficient autoregressive model for nonlinear and nonstationary time series. *IEEE Signal Processing Letters*. 2015;22(7):809-812.
33. Gan M, Chen CLP, Chen L, Zhang CY. Exploiting the interpretability and forecasting ability of the RBF-AR model for nonlinear time series. *International Journal of Systems Science*. 2016;47(8):1868-1876.
34. Cao Y, Lu H, Wen T. A safety computer system based on multi-sensor data processing. *Sensors*. 2019;19(4). <https://doi.org/10.3390/s19040818>
35. Cao Y, Zhang Y, Wen T, Li P. Research on dynamic nonlinear input prediction of fault diagnosis based on fractional differential operator equation in high-speed train control system. *Chaos*. 2019;29(1). Article Number: 013130. <https://doi.org/10.1063/1.5085397>
36. Cao Y, Li P, Zhang Y. Parallel processing algorithm for railway signal fault diagnosis data based on cloud computing. *Future Generation Computer Systems*. 2018;88:279-283.
37. Cao Y, Ma LC, Xiao S, et al. Standard analysis for transfer delay in CTCS-3. *Chinese Journal of Electronics*. 2017;26(5):1057-1063.
38. Cao Y, Wen Y, Meng X, Xu W. Performance evaluation with improved receiver design for asynchronous coordinated multipoint transmissions. *Chinese Journal of Electronics*. 2016;25(2):372-378.



39. Zhang YZ, Cao Y, Wen YH, Liang L, Zou F. Optimization of information interaction protocols in cooperative vehicle-infrastructure systems. *Chinese Journal of Electronics*. 2018;27(2):439-444.
40. Zhang X, Xu L, Ding F, et al. Combined state and parameter estimation for a bilinear state space system with moving average noise. *Journal of the Franklin Institute*. 2018;355(6):3079-3103.
41. Zhang X, Ding F, Alsaadi FE, Hayat T. Recursive parameter identification of the dynamical models for bilinear state space systems. *Nonlinear Dynamics*. 2017;89(4): 2415-2429.
42. Zhang X, Ding F, Xu L, Yang EF. State filtering-based least squares parameter estimation for bilinear systems using the hierarchical identification principle. *IET Control Theory Applications*. 2018;12(12):1704-1713.
43. Zhao SY, Shmaliy YS, Ahn CK, Liu F. Adaptive-horizon iterative UFIR filtering algorithm with applications. *IEEE Transactions on Industrial Electronics*. 2018;65(8):6393-6402.
44. Shi Y, Fang H, Yan M. Kalman filter-based adaptive control for networked systems with unknown parameters and randomly missing outputs. *International Journal of Robust and Nonlinear Control*. 2009;19(18):1976-1992.
45. Kim JH. Note on stability of linear systems with time-varying delay. *Automatica*. 2011;47(9):2118-2121.
46. Liu H, Lu JA, Lü J, Hill DJ. Structure identification of uncertain general complex dynamical networks with time delay. *Automatica*. 2009;45(8):1799-1807.
47. Golub GH, Van Loan CF. *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press; 1996.
48. Yin CC, Zhao JS. Nonexponential asymptotics for the solutions of renewal equations, with applications. *Journal of Applied Probability*. 2008;43(3):815-824.
49. Gao HL, Yin CC. The perturbed sparre Andersen model with a threshold dividend strategy. *Journal of Computational and Applied Mathematics*. 2008;220(1-2):394-408.
50. Yin CC, Yuen KC. Optimality of the threshold dividend strategy for the compound Poisson model. *Statistics & Probability Letters*. 2011;81(12):1841-1846.
51. Yin CC, Wen YZ. Exit problems for jump processes with applications to dividend problems. *Journal of Computational and Applied Mathematics*. 2013;245:30-52.
52. Yin CC, Wen YZ. Optimal dividend problem with a terminal value for spectrally positive Levy processes. *Insurance Mathematics & Economics*. 2013;53(3):769-773.
53. Yin CC, Wang CW. The perturbed compound Poisson risk process with investment and debit interest. *Methodology and Computing in Applied Probability*. 2010;12(3):391-413.
54. Yin CC, Wen YZ, Zhao YX. On the optimal dividend problem for a spectrally positive levy process. *Astin Bulletin*. 2014;44(3):635-651.
55. Yin CC, Zhao JS. Nonexponential asymptotics for the solutions of renewal equations with applications. *Journal of Applied Probability*. 2006;43(3):815-824.
56. Hu YB, Zhou Q, Yu H, Zhou Z, Ding F. Two-stage generalized projection identification algorithms for controlled autoregressive systems. *Circuits Systems and Signal Processing*. 2019;38. <https://doi.org/10.1007/s00034-018-0996-0>
57. Ge ZW, Ding F, Xu L, Alsaedi A, Hayat T. Gradient-based iterative identification method for multivariate equation-error autoregressive moving average systems using the decomposition technique. *Journal of the Franklin Institute*. 2019;356(3):1658-1676.
58. Wan LJ, Ding F. Decomposition- and gradient-based iterative identification algorithms for multivariable systems using the multi-innovation theory. *Circuits Systems and Signal Processing*. 2019;38. <https://doi.org/10.1007/s00034-018-1014-2>
59. Liu QY, Ding F. Auxiliary model-based recursive generalized least squares algorithm for multivariate output-error autoregressive systems using the data filtering. *Circuits Systems and Signal Processing*. 2019;38(2):590-610.
60. Ding JL. The hierarchical iterative identification algorithm for multi-input-output-error systems with autoregressive noise. *Complexity*. 2017;1-11. Article ID 5292894. <https://doi.org/10.1155/2017/5292894>
61. Xu H, Ding F, Yang EF. Modeling a nonlinear process using the exponential autoregressive time series model. *Nonlinear Dynamics*. 2019. <https://doi.org/10.1007/s11071-018-4677-0>
62. Ding F, Liu XP, Liu G. Gradient based and least-squares based iterative identification methods for OE and OEMA systems. *Digital Signal Processing*. 2010;20(3):664-677.