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

Data filtering based least squares iterative algorithm for Hammerstein nonlinear systems by using the model decomposition

Junxia Ma · Feng Ding · Erfu Yang

Received: 9 January 2015 /Revised: /Accepted:

We have used bold font to highlight the changes made in the revised paper.

Abstract This paper focuses on the iterative identification problems for a class of Hammerstein nonlinear systems. By decomposing the system into two fictitious subsystems, a decomposition based least squares iterative algorithm is presented for estimating the parameter vectors in each subsystem. Moreover, a data filtering based decomposition least squares iterative algorithm is proposed. The simulation results indicate that the data filtering based least squares iterative algorithm can generate more accurate parameter estimates than the least squares iterative algorithm.

Keywords Data filtering · Least squares iterative · Model decomposition · Nonlinear system

1 Introduction

The block-oriented nonlinear systems are popular used for modeling and analyzing nonlinear problems in various aspects of our society, such as chemistry processing [1,2], energy harvesting systems [3], signal processing [4–6], predictive control [7,8] and system identification [9,10]. For decades, many approaches have been studied on the system identification and parameter estimation for linear or nonlinear dynamics systems. **The approaches not only can be applied to obtain the mathematical models of the systems [11] but also play an important role in analyzing the controlled dynamics systems [12,13].** For example, Hagenblad et al. derived a maximum likelihood identification method for Wiener models [14]. By the key-term separation principle, Vörös solved the parameter identification problem of nonlinear dynamic systems with both actuator and sensor nonlinearities using three-block cascade models [15]. Based on the least squares principle, Hu et al. derived a recursive extended least squares algorithm for identifying Wiener nonlinear moving average systems [16]. By using the polynomial nonlinear state space approach, Paduart et al. identified a nonliear system with a Wiener–Hammerstein structure [17]. By using the maximum likelihood method, Sun and Liu offered an APSO-aided identification algorithm to identify Hammerstein systems [18].

J.X. Ma · F. Ding (Corresponding author) · E.F. Yang
Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, People's Republic of China
E-mail: fding@jiangnan.edu.cn

J.X. Ma
E-mail: junxia.20@163.com

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

The model decomposition technique can be used to separate a large-scale system into several subsystems with small sizes and to enhance the computational efficiency [19,20]. Recently, Zhang proposed a three-stage least squares iterative identification algorithm for output error moving average systems using the model decomposition [21]; Bai and Liu presented a normalized iterative method to find a least squares solution for a system of bilinear equations by using the decomposition technique [22]; Wang and Ding separated a bilinear-parameter cost function into two linear-parameter cost functions and derived a least squares based and a gradient based iterative identification algorithms for Wiener nonlinear systems [23].

The filtering technique has been proved to be effective in parameter estimation [24,25] and state estimation [26]. Recently, Zhao et al. studied a maximum likelihood method to obtain the parameter estimation of the batch processes by employing the particle filtering approach [27]; Ding et al. used the filtering technique to derive a recursive least squares parameter identification algorithm for systems with colored noise [28]; Wang and Tang presented a filtered three-stage gradient-based iterative algorithm for a class of linear-in-parameters output error autoregressive systems by using the model decomposition and the data filtering technique [29].

By extending the methods in [21,30] from the linear systems to an input nonlinear output error autoregressive (IN-OEAR) system, this paper studies its iterative identification problem. The objective is to decompose a bilinear-parameter system into two fictitious subsystems by using the model decomposition and to present a least squares based iterative algorithm for the IN-OEAR system. Furthermore, using an estimated noise transfer function to filter the input-output data of the system to be identified, a data filtering based least squares iterative algorithm is presented. Compared with the least squares iterative algorithm, the filtering based least squares iterative algorithm can achieve higher estimation accuracy. The proposed algorithms differ from the least squares or gradient based iterative algorithms for Hammerstein nonlinear ARMAX systems using the over-parameterization method in [31] and from the decomposition based iterative least squares algorithm for output error systems based on the partitioned matrix inversion lemma in [32].

Briefly, the rest of this paper is organized as follows. Section 2 gives the identification model of the IN-OEAR systems. Section 3 presents a least squares iterative identification algorithm by using the model decomposition. Section 4 derives a filtering based least squares iterative identification algorithm for the IN-OEAR systems. A numerical example is provided in Section 6 to show the effectiveness of the proposed algorithms. Finally, we give some concluding remarks in Section 7.

2 System description

The typical block-oriented nonlinear models include Hammerstein models (a nonlinear static block followed by a dynamics linear block), Wiener models (a linear dynamics block followed by a static nonlinear block), Hammerstein-Wiener models, and Wiener-Hammerstein models. Here, we consider a Hammerstein nonlinear system with colored noise in Fig. 1,

$$y(t)=\frac{B(z)}{A(z)}\bar{u}(t)+w(t), \tag{1}$$

$$\bar{u}(t)=f(u(t)), \tag{2}$$

where $y(t)$ is the measured output, $w(t)$ is the disturbance with zero mean , $u(t)$ and $\bar{u}(t)$ are the input and output of the nonlinear block, respectively, and $A(z)$ and $B(z)$ are polynomials in the unit backward shift operator z^{-1} ($z^{-1}y(t)=y(t-1)$):

$$A(z):=1+a_1z^{-1}+a_2z^{-2}+\ldots+a_{n_a}z^{-n_a},$$
$$B(z):=b_1z^{-1}+b_2z^{-2}+\ldots+b_{n_b}z^{-n_b}.$$

Assume that the order n_a and n_b are known and $y(t)=0$, $u(t)=0$ and $v(t)=0$ for $t\leqslant 0$. The output of the nonlinear block is a linear combination of the known basic functions $f_j(*)$ and unknown coefficients α_i:

$$\bar{u}(t)=\alpha_1f_1(u(t))+\alpha_2f_2(u(t))+\ldots+\alpha_mf_m(u(t))=\boldsymbol{f}(u(t))\boldsymbol{\alpha},$$

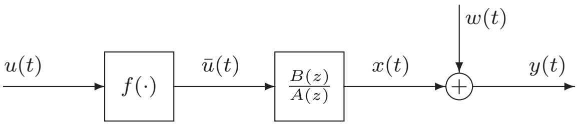


Fig. 1 The Hammerstein nonlinear system with colored noise

where

$$\begin{aligned}\boldsymbol{\alpha} &:= [\alpha_1, \alpha_2, \ldots, \alpha_m]^{\mathrm{T}} \in \mathbb{R}^m,\\ \boldsymbol{f}(u(t)) &:= [f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t))] \in \mathbb{R}^{1\times m}.\end{aligned}$$

The basic functions $f_j(*)$ can be the known order in the input or the trigonometric functions.

For the system with colored noise, the disturbance $w(t)$ can be fitted by an autoregressive process

$$w(t) = \frac{1}{C(z)}v(t), \tag{3}$$

or a moving average process

$$w(t) = D(z)v(t),$$

or an autoregressive moving average process

$$w(t) = \frac{D(z)}{C(z)}v(t),$$

where $v(t)$ is the white noise with zero mean and variances σ^2, $C(z)$ and $D(z)$ are polynomials in the unit backward shift operator z^{-1} [33]:

$$\begin{aligned}C(z) &:= 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_{n_c} z^{-n_c},\\ D(z) &:= 1 + d_1 z^{-1} + d_2 z^{-2} + \ldots + d_{n_d} z^{-n_d}.\end{aligned}$$

This paper assumes the disturbance to be an autoregressive process, and the proposed algorithms can be extended to the other two cases.

Define an intermediate variable:

$$x(t) := \frac{B(z)}{A(z)}\bar{u}(t). \tag{4}$$

Define the parameter vectors and the information vectors as

$$\boldsymbol{a} := \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_a} \end{bmatrix} \in \mathbb{R}^{n_a}, \quad \boldsymbol{b} := \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_b} \end{bmatrix} \in \mathbb{R}^{n_b}, \quad \boldsymbol{c} := \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n_c} \end{bmatrix} \in \mathbb{R}^{n_c},$$

$$\boldsymbol{F}(t) := \begin{bmatrix} \boldsymbol{f}(u(t-1)) \\ \boldsymbol{f}(u(t-2)) \\ \vdots \\ \boldsymbol{f}(u(t-n_b)) \end{bmatrix} \in \mathbb{R}^{n_b\times m},$$

$$\begin{aligned}\boldsymbol{\varphi}_{\mathrm{a}}(t) &:= [-x(t-1), -x(t-2), \ldots, -x(t-n_a)]^{\mathrm{T}} \in \mathbb{R}^{n_a},\\ \boldsymbol{\varphi}_{\mathrm{n}}(t) &:= [-w(t-1), -w(t-2), \ldots, -w(t-n_c)]^{\mathrm{T}} \in \mathbb{R}^{n_c}.\end{aligned}$$

From (3) and (4), we have

$$\begin{aligned}w(t) &= [1 - C(z)]w(t) + v(t)\\ &= -c_1 w(t-1) - c_2 w(t-2) - \ldots - c_{n_c} w(t-n_c) + v(t)\\ &= \boldsymbol{\varphi}_{\mathrm{n}}^{\mathrm{T}}(t)\boldsymbol{c} + v(t), \end{aligned}\tag{5}$$

$$\begin{aligned} x(t) &= [1-A(z)]x(t)+B(z)\bar{u}(t) \\ &= -a_1x(t-1)-a_2x(t-2)-\ldots-a_{n_a}x(t-n_a)+b_1\boldsymbol{f}(u(t-1))\boldsymbol{\alpha}+b_2\boldsymbol{f}(u(t-2))\boldsymbol{\alpha} \\ &\quad +\ldots+b_{n_b}\boldsymbol{f}(u(t-n_b))\boldsymbol{\alpha} \\ &= \boldsymbol{\varphi}_{\mathrm{a}}^{\mathrm{T}}(t)\boldsymbol{a}+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{F}(t)\boldsymbol{\alpha}. \end{aligned} \tag{6}$$

The output $y(t)$ in (1) can be expressed as

$$y(t) = x(t)+w(t) \tag{7}$$

$$\quad = \boldsymbol{\varphi}_{\mathrm{a}}^{\mathrm{T}}(t)\boldsymbol{a}+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{F}(t)\boldsymbol{\alpha}+\boldsymbol{\varphi}_{\mathrm{n}}^{\mathrm{T}}(t)\boldsymbol{c}+v(t). \tag{8}$$

This is the identification model for the Hammerstein nonlinear system.

3 The decomposition based least squares iterative algorithm

It is worth pointing out that model (8) contains the product of the parameters $\boldsymbol{b}$ of the linear part and $\boldsymbol{\alpha}$ of the nonlinear part. The pair $\beta\boldsymbol{b}$ and $\boldsymbol{\alpha}/\beta$ leads to the same input-output relation for any nonzero constant β. In order to ensure identifiability, we assume that $\|\boldsymbol{\alpha}\|=1$ and the first entry of the vector $\boldsymbol{\alpha}$ is positive, i.e., $\alpha_1>0$. Although we can use the Kronecker product to transform the bilinear parameter identification problem to a linear parameter identification problem [34,35], the dimension of the resulting unknown parameter vector increases, so does the calculation load. Here, we decompose this system into two fictitious subsystems: one containing the parameter vector $\boldsymbol{\theta}:=\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{b}\end{bmatrix}$, and the other containing the parameter vector $\boldsymbol{\vartheta}:=\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{c}\end{bmatrix}$. Let $k=1,2,3,\ldots$ be an iterative variable, $\hat{\boldsymbol{\theta}}_k(t):=\begin{bmatrix}\hat{\boldsymbol{a}}_k(t)\\ \hat{\boldsymbol{b}}_k(t)\end{bmatrix}$ and $\hat{\boldsymbol{\vartheta}}_k(t):=\begin{bmatrix}\hat{\boldsymbol{\alpha}}_k(t)\\ \hat{\boldsymbol{c}}_k(t)\end{bmatrix}$ be the estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ at iteration k. Define two fictitious outputs:

$$\begin{aligned} y_1(t) &:= y(t)-\boldsymbol{\varphi}_{\mathrm{n}}^{\mathrm{T}}(t)\boldsymbol{c} \\ &= \boldsymbol{\varphi}_{\mathrm{a}}^{\mathrm{T}}(t)\boldsymbol{a}+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{F}(t)\boldsymbol{\alpha}+v(t), \\ &= \boldsymbol{\varphi}_1^{\mathrm{T}}(t)\boldsymbol{\theta}+v(t), \end{aligned} \tag{9}$$

$$\begin{aligned} y_2(t) &:= y(t)-\boldsymbol{\varphi}_{\mathrm{a}}^{\mathrm{T}}(t)\boldsymbol{a} \\ &= \boldsymbol{b}^{\mathrm{T}}\boldsymbol{F}(t)\boldsymbol{\alpha}+\boldsymbol{\varphi}_{\mathrm{n}}^{\mathrm{T}}(t)\boldsymbol{c}+v(t), \\ &= \boldsymbol{\varphi}_2^{\mathrm{T}}(t)\boldsymbol{\vartheta}+v(t), \end{aligned} \tag{10}$$

where

$$\boldsymbol{\varphi}_1(t):=\begin{bmatrix}\boldsymbol{\varphi}_{\mathrm{a}}(t)\\ \boldsymbol{F}(t)\boldsymbol{\alpha}\end{bmatrix}\in\mathbb{R}^{n_a+n_b},$$

$$\boldsymbol{\varphi}_2(t):=\begin{bmatrix}\boldsymbol{F}^{\mathrm{T}}(t)\boldsymbol{b}\\ \boldsymbol{\varphi}_{\mathrm{n}}(t)\end{bmatrix}\in\mathbb{R}^{m+n_c}.$$

Opt a set of data from $j=t-L+1$ to $j=t$ (L denotes the data length) and define two quadratic criterion functions:

$$J_1(\boldsymbol{\theta})=\sum_{j=t-L+1}^{t}[y_1(j)-\boldsymbol{\varphi}_1^{\mathrm{T}}(j)\boldsymbol{\theta}]^2,$$

$$J_2(\boldsymbol{\vartheta})=\sum_{j=t-L+1}^{t}[y_2(j)-\boldsymbol{\varphi}_2^{\mathrm{T}}(j)\boldsymbol{\vartheta}]^2.$$

Based on the least squares principle, letting the partial derivative of $J_1(\boldsymbol{\theta})$ and $J_2(\boldsymbol{\vartheta})$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ be zero, respectively, we can obtain the following least squares iterative algorithm:

$$\hat{\boldsymbol{\theta}}_k(t)=\left[\sum_{j=t-L+1}^{t}\boldsymbol{\varphi}_1(j)\boldsymbol{\varphi}_1^{\mathrm{T}}(j)\right]^{-1}\sum_{j=t-L+1}^{t}\boldsymbol{\varphi}_1(j)y_1(j), \tag{11}$$

$$\hat{\boldsymbol{\vartheta}}_k(t) = \left[\sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_2(j)\boldsymbol{\varphi}_2^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_2(j)y_2(j). \tag{12}$$

Substituting (9) into (11) and (10) into (12) gives

$$\hat{\boldsymbol{\theta}}_k(t) = \left[\sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_1(j)\boldsymbol{\varphi}_1^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_1(j)[y(j) - \boldsymbol{\varphi}_{\mathrm{n}}^{\mathrm{T}}(j)\boldsymbol{c}], \tag{13}$$

$$\hat{\boldsymbol{\vartheta}}_k(t) = \left[\sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_2(j)\boldsymbol{\varphi}_2^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_2(j)[y(j) - \boldsymbol{\varphi}_{\mathrm{a}}^{\mathrm{T}}(j)\boldsymbol{a}]. \tag{14}$$

The difficulty is that the right-hand sides of (13) and (14) contain the unknown parameter vectors $\boldsymbol{c}$ and $\boldsymbol{a}$, the information vectors $\boldsymbol{\varphi}_1(t)$ and $\boldsymbol{\varphi}_2(t)$ contain the unknown parameter vectors $\boldsymbol{\alpha}$ and $\boldsymbol{b}$ and the unknown intermediate variables $x(t-i)$ and $w(t-i)$, so it is impossible to compute $\hat{\boldsymbol{\theta}}_k(t)$ and $\hat{\boldsymbol{\vartheta}}_k(t)$ by (13) and (14) directly. Here, the solution is based on the hierarchical identification principle [36]. Let $\hat{w}_k(t-i)$ and $\hat{x}_k(t-i)$ be the estimates of $w(t-i)$ and $x(t-i)$ at iteration k, $\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t)$, $\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t)$, $\hat{\boldsymbol{\varphi}}_{1,k}(t)$, and $\hat{\boldsymbol{\varphi}}_{2,k}(t)$ be the estimates of $\boldsymbol{\varphi}_{\mathrm{a}}(t)$, $\boldsymbol{\varphi}_{\mathrm{n}}(t)$, $\boldsymbol{\varphi}_1(t)$, and $\boldsymbol{\varphi}_2(t)$ at iteration k and define

$$\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t) := [-\hat{x}_{k-1}(t-1), -\hat{x}_{k-1}(t-2), \ldots, -\hat{x}_{k-1}(t-n_a)]^{\mathrm{T}} \in \mathbb{R}^{n_a},$$
$$\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t) := [-\hat{w}_{k-1}(t-1), -\hat{w}_{k-1}(t-2), \ldots, -\hat{w}_{k-1}(t-n_c)]^{\mathrm{T}} \in \mathbb{R}^{n_c},$$
$$\hat{\boldsymbol{\varphi}}_{1,k}(t) := \begin{bmatrix} \hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t) \\ \boldsymbol{F}(t)\hat{\boldsymbol{\alpha}}_{k-1}(t) \end{bmatrix} \in \mathbb{R}^{n_a+n_b},$$
$$\hat{\boldsymbol{\varphi}}_{2,k}(t) := \begin{bmatrix} \boldsymbol{F}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}_k(t) \\ \hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t) \end{bmatrix} \in \mathbb{R}^{m+n_c}.$$

From (6), we have $x_k(t-i) = \boldsymbol{\varphi}_{\mathrm{a}}^{\mathrm{T}}(t-i)\boldsymbol{a} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{F}(t-i)\boldsymbol{\alpha}$. Replacing $\boldsymbol{\varphi}_{\mathrm{a}}(t-i)$, $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{\alpha}$ with their estimates $\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t-i)$, $\hat{\boldsymbol{a}}_k(t)$, $\hat{\boldsymbol{b}}_k(t)$ and $\hat{\boldsymbol{\alpha}}_k(t)$ gives

$$\hat{x}_k(t-i) = \hat{\boldsymbol{\varphi}}_{\mathrm{a},k}^{\mathrm{T}}(t-i)\hat{\boldsymbol{a}}_k(t) + \hat{\boldsymbol{b}}_k^{\mathrm{T}}(t)\boldsymbol{F}(t-i)\hat{\boldsymbol{\alpha}}_k(t).$$

From (7), we have $w(t-i) = y(t-i) - x(t-i)$. Replacing $x(t-i)$ with $\hat{x}_k(t-i)$, we can compute the estimate of $w(t)$ through:

$$\hat{w}_k(t-i) = y(t-i) - \hat{x}_k(t-i).$$

Replacing the unknown $\boldsymbol{c}$ and $\boldsymbol{\varphi}_1(t)$ in (13) with their estimates $\hat{\boldsymbol{c}}_{k-1}(t)$ and $\hat{\boldsymbol{\varphi}}_{1,k}(t)$, the unknown $\boldsymbol{a}$, $\boldsymbol{\varphi}_2(t)$ and $\boldsymbol{\varphi}_{\mathrm{a}}(t)$ in (14) with their estimates $\hat{\boldsymbol{a}}_k(t)$, $\hat{\boldsymbol{\varphi}}_{2,k}(t)$ and $\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t)$, we can summarize the decomposition based least squares iterative (D-LSI) algorithm for estimating $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ as follows:

$$\hat{\boldsymbol{\theta}}_k(t) = \left[\sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{1,k}(j)\hat{\boldsymbol{\varphi}}_{1,k}^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{1,k}(j)[y(j) - \hat{\boldsymbol{\varphi}}_{\mathrm{n},k}^{\mathrm{T}}(j)\hat{\boldsymbol{c}}_{k-1}(j)], \tag{15}$$

$$\hat{\boldsymbol{\vartheta}}_k(t) = \left[\sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{2,k}(j)\hat{\boldsymbol{\varphi}}_{2,k}^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{2,k}(j)[y(j) - \hat{\boldsymbol{\varphi}}_{\mathrm{a},k}^{\mathrm{T}}(j)\hat{\boldsymbol{a}}_k(j)], \tag{16}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t) = [-\hat{x}_{k-1}(t-1), -\hat{x}_{k-1}(t-2), \ldots, -\hat{x}_{k-1}(t-n_a)]^{\mathrm{T}}, \tag{17}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t) = [-\hat{w}_{k-1}(t-1), -\hat{w}_{k-1}(t-2), \ldots, -\hat{w}_{k-1}(t-n_c)]^{\mathrm{T}}, \tag{18}$$

$$\hat{\boldsymbol{\varphi}}_{1,k}(t) = [\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}^{\mathrm{T}}(t), \hat{\boldsymbol{\alpha}}_{k-1}^{\mathrm{T}}(t)\boldsymbol{F}^{\mathrm{T}}(t)]^{\mathrm{T}}, \tag{19}$$

$$\hat{\boldsymbol{\varphi}}_{2,k}(t) = [\hat{\boldsymbol{b}}_k^{\mathrm{T}}(t)\boldsymbol{F}(t), \hat{\boldsymbol{\varphi}}_{\mathrm{n},k}^{\mathrm{T}}(t)]^{\mathrm{T}}, \tag{20}$$

$$\hat{x}_k(t-i) = \hat{\boldsymbol{\varphi}}_{\mathrm{a},k}^{\mathrm{T}}(t-i)\hat{\boldsymbol{a}}_k(t) + \hat{\boldsymbol{b}}_k^{\mathrm{T}}(t)\boldsymbol{F}(t-i)\hat{\boldsymbol{\alpha}}_k(t), \quad i = 1, 2, \ldots, n_a \tag{21}$$

$$\hat{w}_k(t-j) = y(t-j) - \hat{x}_k(t-j), \quad j = 1, 2, \ldots, n_c \tag{22}$$

$$\boldsymbol{F}(t)=\left[\begin{array}{cccc} f_1(u(t-1)) & f_2(u(t-1)) & \ldots & f_m(u(t-1)) \\ f_1(u(t-2)) & f_2(u(t-2)) & \ldots & f_m(u(t-2)) \\ \vdots & \vdots & & \vdots \\ f_1(u(t-n_b)) & f_2(u(t-n_b)) & \ldots & f_m(u(t-n_b)) \end{array}\right], \tag{23}$$

$$\hat{\boldsymbol{a}}_k(t)=\hat{\boldsymbol{\theta}}_k(t)(1:n_a), \tag{24}$$

$$\hat{\boldsymbol{b}}_k(t)=\hat{\boldsymbol{\theta}}_k(t)(n_a+1:n_a+n_b), \tag{25}$$

$$\hat{\boldsymbol{\alpha}}_k(t)=\operatorname{sgn}[\hat{\boldsymbol{\vartheta}}_k(t)(1)]\frac{\hat{\boldsymbol{\vartheta}}_k(t)(1:m)}{\|\hat{\boldsymbol{\vartheta}}_k(t)(1:m)\|}. \tag{26}$$

To initialize the D-LSI algorithm, the initial value $\hat{\boldsymbol{\theta}}_0(t)=\begin{bmatrix}\hat{\boldsymbol{a}}_0(t)\\ \hat{\boldsymbol{b}}_0(t)\end{bmatrix}$ is generally taken to be a nonzero vector with $\hat{\boldsymbol{b}}_0(t)\neq 0$, $\hat{\boldsymbol{\alpha}}_0(t)$ is taken to be a vector with $\|\hat{\boldsymbol{\alpha}}_0(t)\|=1$, and $\hat{\boldsymbol{c}}_0(t)$ is taken to be an arbitrary real vector. The initial value of intermediate variables $\hat{w}_0(t-i)$ and $\hat{x}_0(t-i)$ are taken to be two random numbers.

4 The filtering-based least squares iterative algorithm

Using the polynomial $C(z)$ (a linear filter) to filter the input-output data, the model in (1) can be transformed into two identification models: an input nonlinear output error model with white noise and an autoregressive noise model. Multiplying both sides of Equation (1) by $C(z)$ yields

$$C(z)y(t)=\frac{B(z)}{A(z)}C(z)\bar{u}(t)+v(t). \tag{27}$$

Define the filtered output $y_{\mathrm{f}}(t)$ and input $\bar{u}_{\mathrm{f}}(t)$:

$$\begin{aligned} y_{\mathrm{f}}(t) &:= C(z)y(t)\\ &= y(t)+c_1y(t-1)+c_2y(t-2)+\ldots+c_ny(t-n),\\ \bar{u}_{\mathrm{f}}(t) &:= C(z)\bar{u}(t)\\ &= C(z)[\alpha_1f_1(u(t))+\alpha_2f_2(u(t))+\ldots+\alpha_mf_m(u(t))]\\ &= \alpha_1g_1(t)+\alpha_2g_2(t)+\ldots+\alpha_mg_m(t), \end{aligned}$$

where

$$g_j(t):=C(z)f_j(u(t)),\quad j=1,2,\ldots,m.$$

Define an information matrix:

$$\boldsymbol{G}(t):=\left[\begin{array}{cccc} g_1(t-1) & g_2(t-1) & \ldots & g_m(t-1) \\ g_1(t-2) & g_2(t-2) & \ldots & g_m(t-2) \\ \vdots & \vdots & & \vdots \\ g_1(t-n_b) & g_2(t-n_b) & \ldots & g_m(t-n_b) \end{array}\right]\in\mathbb{R}^{n_b\times m}.$$

Then Equation (27) can be rewritten as

$$y_{\mathrm{f}}(t)=\frac{B(z)}{A(z)}\bar{u}_{\mathrm{f}}(t)+v(t).$$

Define an intermediate variable:

$$x_{\mathrm{f}}(t):=\frac{B(z)}{A(z)}\bar{u}_{\mathrm{f}}(t).$$

Then we have

$$\begin{aligned} x_{\mathrm{f}}(t) &= [1-A(z)]x_{\mathrm{f}}(t)+B(z)\bar{u}_{\mathrm{f}}(t)\\ &= \boldsymbol{\varphi}_{\mathrm{f}}^{\mathrm{T}}(t)\boldsymbol{a}+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{G}(t)\boldsymbol{\alpha}, \end{aligned} \tag{28}$$

where

$$\boldsymbol{\varphi}_{\rm f}(t) := [-x_{\rm f}(t-1), -x_{\rm f}(t-2), \ldots, -x_{\rm f}(t-n_a)]^{\rm T} \in \mathbb{R}^{n_a}.$$

The filtered output $y_{\rm f}(t)$ can be expressed as

$$\begin{aligned} y_{\rm f}(t) &= x_{\rm f}(t) + v(t)\\ &= \boldsymbol{\varphi}_{\rm f}^{\rm T}(t)\boldsymbol{a} + \boldsymbol{b}^{\rm T}\boldsymbol{G}(t)\boldsymbol{\alpha} + v(t). \end{aligned} \tag{29}$$

Define a fictitious output and two quadratic criterion functions as

$$\begin{aligned} y_3(t) &:= y_{\rm f}(t) - \boldsymbol{\varphi}_{\rm f}^{\rm T}(t)\boldsymbol{a}\\ &= \boldsymbol{b}^{\rm T}\boldsymbol{G}(t)\boldsymbol{\alpha} + v(t),\\ J_3(\boldsymbol{\theta}) &:= \sum_{j=t-L+1}^{t} [y_{\rm f}(j) - \boldsymbol{\varphi}_3^{\rm T}(j)\boldsymbol{\theta}]^2,\\ J_4(\boldsymbol{\alpha}) &:= \sum_{j=t-L+1}^{t} [y_3(j) - \boldsymbol{\varphi}_4^{\rm T}(j)\boldsymbol{\alpha}]^2, \end{aligned}$$

where

$$\boldsymbol{\varphi}_3(t) := \begin{bmatrix} \boldsymbol{\varphi}_{\rm f}(t)\\ \boldsymbol{G}(t)\boldsymbol{\alpha} \end{bmatrix} \in \mathbb{R}^{n_a+n_b}, \tag{30}$$

$$\boldsymbol{\varphi}_4(t) := \boldsymbol{G}^{\rm T}(t)\boldsymbol{b} \in \mathbb{R}^{m}. \tag{31}$$

Minimizing the criterion functions $J_3(\boldsymbol{\theta})$ and $J_4(\boldsymbol{\alpha})$, and letting the partial derivatives of $J_3(\boldsymbol{\theta})$ and $J_4(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ be zero, respectively, give the following iterative algorithm to estimate $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$:

$$\hat{\boldsymbol{\theta}}_k(t) = \left[\sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_3(j)\boldsymbol{\varphi}_3^{\rm T}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_3(j) y_{\rm f}(j), \tag{32}$$

$$\begin{aligned} \hat{\boldsymbol{\alpha}}_k(t) &= \left[\sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_4(j)\boldsymbol{\varphi}_4^{\rm T}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_4(j) y_3(j)\\ &= \left[\sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_4(j)\boldsymbol{\varphi}_4^{\rm T}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_4(j)[y_{\rm f}(j) - \boldsymbol{\varphi}_{\rm f}^{\rm T}(j)\boldsymbol{a}]. \end{aligned} \tag{33}$$

However, the polynomial $C(z)$ is unknown, so are the filtered output $y_{\rm f}(t)$, the filtered input $\bar{u}_{\rm f}(t)$, and the filtered information matrix $\boldsymbol{G}(t)$. Thus it is impossible to obtain the estimates $\hat{\boldsymbol{\theta}}_k(t)$ and $\hat{\boldsymbol{\alpha}}_k(t)$ by (32) and (33). Here, we need to compute the parameter estimation vector $\hat{\boldsymbol{c}}_k(t) = [\hat{c}_{1,k}(t), \hat{c}_{2,k}(t), \ldots, \hat{c}_{n_c,k}(t)]^{\rm T}$ firstly, and then use the estimated polynomial $\hat{C}_k(t,z) := 1 + \hat{c}_{1,k}(t)z^{-1} + \hat{c}_{2,k}(t)z^{-2} + \ldots + \hat{c}_{n_c,k}(t)z^{-n_c}$ to filter $y(t)$, $\bar{u}(t)$ to obtain the estimates $\hat{y}_{{\rm f},k}(t)$, $\hat{\bar{u}}_{{\rm f},k}(t)$ and $\hat{\boldsymbol{G}}_k(t)$.

According to (5), define a quadratic criterion function:

$$J_5(\boldsymbol{c}) := \sum_{j=t-L+1}^{t} [w(j) - \boldsymbol{\varphi}_{\rm n}(j)\boldsymbol{c}]^2.$$

Minimizing the criterion functions $J_5(\boldsymbol{c})$ gives the iterative estimate of $\boldsymbol{c}$:

$$\hat{\boldsymbol{c}}_k(t) = \left[\sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_{\rm n}(j)\boldsymbol{\varphi}_{\rm n}^{\rm T}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \boldsymbol{\varphi}_{\rm n}(j) w(j). \tag{34}$$

We can find that the right-hand side of (34) contains the unknown information vector $\boldsymbol{\varphi}_{\rm n}(t)$ and intermediate variable $w(t)$. Similarly, replacing the unknown $\boldsymbol{\varphi}_{\rm n}(t)$ and $w(t)$ in (34) with their corresponding estimates $\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t)$ and $\hat{w}_k(t)$, we can obtain the least squares iterative algorithm for computing the estimate $\hat{\boldsymbol{c}}_k(t)$ as follows:

$$\hat{\boldsymbol{c}}_k(t) = \left[\sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(j)\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(j)\hat{w}_k(j), \tag{35}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t) = [-\hat{w}_{k-1}(t-1), -\hat{w}_{k-1}(t-2), \ldots, -\hat{w}_{k-1}(t-n_c)]^{\mathrm{T}}, \tag{36}$$

$$\hat{w}_k(t-i) = y(t-i) - \hat{\boldsymbol{\varphi}}_{\mathrm{a},k}^{\mathrm{T}}(t-i)\hat{\boldsymbol{a}}_{k-1}(t) - \hat{\boldsymbol{b}}_{k-1}^{\mathrm{T}}\boldsymbol{F}(t-i)\hat{\boldsymbol{\alpha}}_{k-1}(t). \tag{37}$$

Using the obtained estimate $\hat{\boldsymbol{c}}_k(t)$ to construct the polynomial

$$\hat{C}_k(t,z) = 1 + \hat{c}_{1,k}(t)z^{-1} + \hat{c}_{2,k}(t)z^{-2} + \ldots + \hat{c}_{n_c,k}(t)z^{-n_c}$$

to filter $y(t)$ and $\hat{u}(t)$ gives the filtered estimates $\hat{y}_{\mathrm{f},k}(t)$ and $\hat{u}_{\mathrm{f},k}(t)$:

$$\begin{aligned}\hat{y}_{\mathrm{f},k}(t) &= \hat{C}_k(t,z)y(t)\\ &= y(t) + \hat{c}_{1,k}(t)y(t-1) + \hat{c}_{2,k}(t)y(t-2) + \ldots + \hat{c}_{n_c,k}(t)y(t-n_c),\\ \hat{u}_{\mathrm{f},k}(t) &= \hat{C}_k(t,z)\hat{u}(t)\\ &= \hat{C}_k(t,z)[\hat{\alpha}_{1,k}(t)f_1(u(t)) + \hat{\alpha}_{2,k}(t)f_2(u(t)) + \ldots + \hat{\alpha}_{m,k}(t)f_m(u(t))]\\ &= \hat{\alpha}_1(t)\hat{g}_{1,k}(t) + \hat{\alpha}_{2,k}(t)\hat{g}_{2,k}(t) + \ldots + \hat{\alpha}_{m,k}(t)\hat{g}_{m,k}(t),\end{aligned}$$

where $\hat{g}_{j,k}(t)$ can be computed by

$$\begin{aligned}\hat{g}_{j,k}(t) &= \hat{C}_k(t,z)f_j(u(t))\\ &= f_j(u(t)) + \hat{c}_{1,k}(t)f_j(u(t-1)) + \hat{c}_{2,k}(t)f_j(u(t-2)) + \ldots + \hat{c}_{n_c,k}(t)f_j(u(t-n_c)).\end{aligned}$$

Let $\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t)$ be the estimate of $\boldsymbol{\varphi}_{\mathrm{f}}(t)$ and define

$$\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t) := [-\hat{x}_{\mathrm{f},k-1}(t-1), -\hat{x}_{\mathrm{f},k-1}(t-2), \ldots, -\hat{x}_{\mathrm{f},k-1}(t-n_a)] \in \mathbb{R}^{n_a}.$$

From (28), we have $x_{\mathrm{f}}(t-i) = \boldsymbol{\varphi}_{\mathrm{f}}^{\mathrm{T}}(t-i)\boldsymbol{a} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{G}(t-i)\boldsymbol{\alpha}$. Replacing the parameter vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{\alpha}$ with their estimates $\hat{\boldsymbol{a}}_k(t)$, $\hat{\boldsymbol{b}}_k(t)$ and $\hat{\boldsymbol{\alpha}}_k(t)$ at iteration k and the unknown $\boldsymbol{\varphi}_{\mathrm{f}}(t-i)$ and $\boldsymbol{G}(t-i)$ with their estimates $\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t-i)$ and $\hat{\boldsymbol{G}}_k(t-i)$, respectively, the estimate $\hat{x}_{\mathrm{f},k}(t-i)$ can be computed by

$$\hat{x}_{\mathrm{f},k}(t-i) = \hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t-i)\hat{\boldsymbol{a}}_k(t) + \hat{\boldsymbol{b}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{G}}_k(t-i)\hat{\boldsymbol{\alpha}}_k(t).$$

According to (30) and (31), we define

$$\hat{\boldsymbol{\varphi}}_{3,k}(t) := \begin{bmatrix}\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t)\\ \hat{\boldsymbol{G}}_k(t)\hat{\boldsymbol{\alpha}}_{k-1}(t)\end{bmatrix} \in \mathbb{R}^{n_a+n_b},$$

$$\hat{\boldsymbol{\varphi}}_{4,k}(t) := \hat{\boldsymbol{G}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{b}}_k(t) \in \mathbb{R}^{m}.$$

Replacing $\boldsymbol{\varphi}_3(t)$ and $y_{\mathrm{f}}(t)$ in (32) with their estimates $\hat{\boldsymbol{\varphi}}_{3,k}(t)$ and $\hat{y}_{\mathrm{f}}(t)$, replacing $\boldsymbol{\varphi}_4(t)$, $y_{\mathrm{f}}(t)$ and $\boldsymbol{\varphi}_{\mathrm{f}}(t)$ in (33) with their estimates $\hat{\boldsymbol{\varphi}}_{4,k}(t)$, $\hat{y}_{\mathrm{f}}(t)$ and $\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t)$, respectively, we can obtain the following data filtering based least squares iterative algorithm by using the model decomposition technique (the F-D-LSI algorithm for short):

$$\hat{\boldsymbol{\theta}}_k(t) = \left[\sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{3,k}(j)\hat{\boldsymbol{\varphi}}_{3,k}^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{3,k}(j)\hat{y}_{\mathrm{f},k}(j), \tag{38}$$

$$\hat{\boldsymbol{\varphi}}_{3,k}(t) = [\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}^{\mathrm{T}}(t), \hat{\boldsymbol{\alpha}}_{k-1}^{\mathrm{T}}(t)\hat{\boldsymbol{G}}_k^{\mathrm{T}}(t)]^{\mathrm{T}}, \tag{39}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t) = [-\hat{x}_{\mathrm{f},k-1}(t-1), -\hat{x}_{\mathrm{f},k-1}(t-2), \ldots, -\hat{x}_{\mathrm{f},k-1}(t-n_a)]^{\mathrm{T}}, \quad i = 1, 2, \ldots, n_a, \tag{40}$$

$$\hat{x}_{\mathrm{f},k}(t-i) = \hat{\boldsymbol{\varphi}}_{\mathrm{f},k}^{\mathrm{T}}(t-i)\hat{\boldsymbol{a}}_k(t) + \hat{\boldsymbol{b}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{G}}_k(t-i)\hat{\boldsymbol{\alpha}}_k(t), \tag{41}$$

$$\hat{\boldsymbol{\alpha}}_k(t) = \left[\sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{4,k}(j)\hat{\boldsymbol{\varphi}}_{4,k}^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{4,k}(j)[\hat{y}_{\mathrm{f},k}(j) - \hat{\boldsymbol{\varphi}}_{\mathrm{f},k}^{\mathrm{T}}(j)\hat{\boldsymbol{a}}_k(j)], \tag{42}$$

$$\hat{\boldsymbol{\varphi}}_{4,k}(t) = \hat{\boldsymbol{G}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{b}}_k(t), \tag{43}$$

$$\hat{\boldsymbol{G}}_k(t) = \left[\begin{array}{cccc} \hat{g}_{1,k}(t-1) & \hat{g}_{2,k}(t-1) & \dots & \hat{g}_{m,k}(t-1) \\ \hat{g}_{1,k}(t-2) & \hat{g}_{2,k}(t-2) & \dots & \hat{g}_{m,k}(t-2) \\ \vdots & \vdots & & \vdots \\ \hat{g}_{1,k}(t-n_b) & \hat{g}_{2,k}(t-n_b) & \dots & \hat{g}_{m,k}(t-n_b) \end{array}\right], \tag{44}$$

$$\hat{g}_{j,k}(t) = f_j(u(t)) + \hat{c}_{1,k}(t)f_j(u(t-1)) + \hat{c}_{2,k}(t)f_j(u(t-2)) + \ldots + \hat{c}_{n_c,k}(t)f_j(u(t-n_c)), \tag{45}$$

$$\hat{y}_{\mathrm{f},k}(t) = y(t) + \hat{c}_{1,k}(t)y(t-1) + \hat{c}_{2,k}(t)y(t-2) + \ldots + \hat{c}_{n_c,k}(t)y(t-n_c), \tag{46}$$

$$\hat{\boldsymbol{c}}_k(t) = \left[\sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(j)\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}^{\mathrm{T}}(j)\right]^{-1} \sum_{j=t-L+1}^{t} \hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(j)\hat{w}_k(j), \tag{47}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t) = [-\hat{w}_{k-1}(t-1), -\hat{w}_{k-1}(t-2), \ldots, -\hat{w}_{k-1}(t-n_c)]^{\mathrm{T}}, \tag{48}$$

$$\hat{w}_k(t-j) = y(t-j) - \hat{\boldsymbol{\varphi}}_{\mathrm{a},k}^{\mathrm{T}}(t-j)\hat{\boldsymbol{a}}_{k-1}(t) - \hat{\boldsymbol{b}}_{k-1}^{\mathrm{T}}(t)\boldsymbol{F}(t-j)\hat{\boldsymbol{\alpha}}_{k-1}(t), \quad j = 1, 2, \ldots, n_c \tag{49}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t) = [-\hat{x}_{k-1}(t-1), -\hat{x}_{k-1}(t-2), \ldots, -\hat{x}_{k-1}(t-n_a)]^{\mathrm{T}}, \tag{50}$$

$$\hat{x}_k(t-i) = [\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}^{\mathrm{T}}(t-i), \hat{\boldsymbol{\alpha}}_k^{\mathrm{T}}(t)\boldsymbol{F}^{\mathrm{T}}(t-i)]\hat{\boldsymbol{\theta}}_k(t), \quad i = 1, 2, \ldots, n_a, \tag{51}$$

$$\boldsymbol{F}(t) = \left[\begin{array}{cccc} f_1(u(t-1)) & f_2(u(t-1)) & \dots & f_m(u(t-1)) \\ f_1(u(t-2)) & f_2(u(t-2)) & \dots & f_m(u(t-2)) \\ \vdots & \vdots & & \vdots \\ f_1(u(t-n_b)) & f_2(u(t-n_b)) & \dots & f_m(u(t-n_b)) \end{array}\right], \tag{52}$$

$$\hat{\boldsymbol{a}}_k(t) = \hat{\boldsymbol{\theta}}_k(t)(1 : n_a), \tag{53}$$

$$\hat{\boldsymbol{b}}_k(t) = \hat{\boldsymbol{\theta}}_k(t)(na+1 : n_a+n_b), \tag{54}$$

$$\bar{\boldsymbol{\alpha}}_k(t) = \mathrm{sgn}[\hat{\boldsymbol{\alpha}}_k(t)(1)]\frac{\hat{\boldsymbol{\alpha}}_k(t)}{\|\hat{\boldsymbol{\alpha}}_k(t)\|}, \quad \hat{\boldsymbol{\alpha}}_k(t) = \bar{\boldsymbol{\alpha}}_k(t), \tag{55}$$

$$\hat{\boldsymbol{c}}_k(t) = [\hat{c}_{1,k}(t), \hat{c}_{2,k}(t), \ldots, \hat{c}_{n_c,k}(t)]^{\mathrm{T}}, \tag{56}$$

$$\hat{\boldsymbol{\Theta}}_k(t) = [\hat{\boldsymbol{\theta}}_k^{\mathrm{T}}(t), \hat{\boldsymbol{\alpha}}_k^{\mathrm{T}}(t), \hat{\boldsymbol{c}}_k^{\mathrm{T}}(t)]^{\mathrm{T}}. \tag{57}$$

To initialize the F-D-LSI algorithm: let $k = 1$, and set the initial values: $\hat{\boldsymbol{\theta}}_0(t) = \begin{bmatrix}\hat{\boldsymbol{a}}_0(t)\\ \hat{\boldsymbol{b}}_0(t)\end{bmatrix}$ be any nonzero real vector with $\hat{\boldsymbol{b}}_0(t) \neq 0$, $\hat{\boldsymbol{\alpha}}_0(t)$ be an real vector with $\|\hat{\boldsymbol{\alpha}}_0(t)\| = 1$, $\hat{\boldsymbol{c}}_0(t)$ be an arbitrary real vector, $\hat{x}_{\mathrm{f},0}(t-i)$, $\hat{x}_0(t-i)$ and $\hat{w}_0(t-i)$ are random numbers, $\hat{y}_{\mathrm{f},0}(t-i) = 1/p_0$, p_0 is taken to be a large number, for example $p_0 = 10^6$. The flowchart of the F-D-LSI algorithm for computing $\hat{\boldsymbol{\theta}}_k(t)$, $\hat{\boldsymbol{\alpha}}_k(t)$ and $\hat{\boldsymbol{c}}_k(t)$ is shown in Fig. 2.

5 The F-D-LSI algorithm with finite measurement data

On the basis of the F-D-LSI algorithm, this section simply gives the data filtering based least squares iterative algorithm with finite measurement data. Letting $t = L$, from $J_3(\boldsymbol{\theta})$, $J_4(\boldsymbol{\alpha})$ and $J_5(\boldsymbol{c})$, we have

$$J_6(\boldsymbol{\theta}) := \sum_{j=1}^{L}[y_{\mathrm{f}}(j) - \boldsymbol{\varphi}_3(j)\boldsymbol{\theta}]^2,$$

$$J_7(\boldsymbol{\alpha}) := \sum_{j=1}^{L}[y_{\mathrm{f}}(t) - \boldsymbol{\varphi}_{\mathrm{f}}^{\mathrm{T}}(t)\boldsymbol{a} - \boldsymbol{\varphi}_4^{\mathrm{T}}(j)\boldsymbol{\alpha}]^2,$$

$$J_8(\boldsymbol{c}) := \sum_{j=1}^{L}[w(j) - \boldsymbol{\varphi}_{\mathrm{n}}(j)\boldsymbol{c}]^2.$$

Applying the similar way of deriving the F-D-LSI algorithm and minimizing the criterion functions $J_6(\boldsymbol{\theta})$, $J_7(\boldsymbol{\alpha})$ and $J_8(\boldsymbol{c})$, we can obtain the F-D-LSI algorithm with finite measurement data

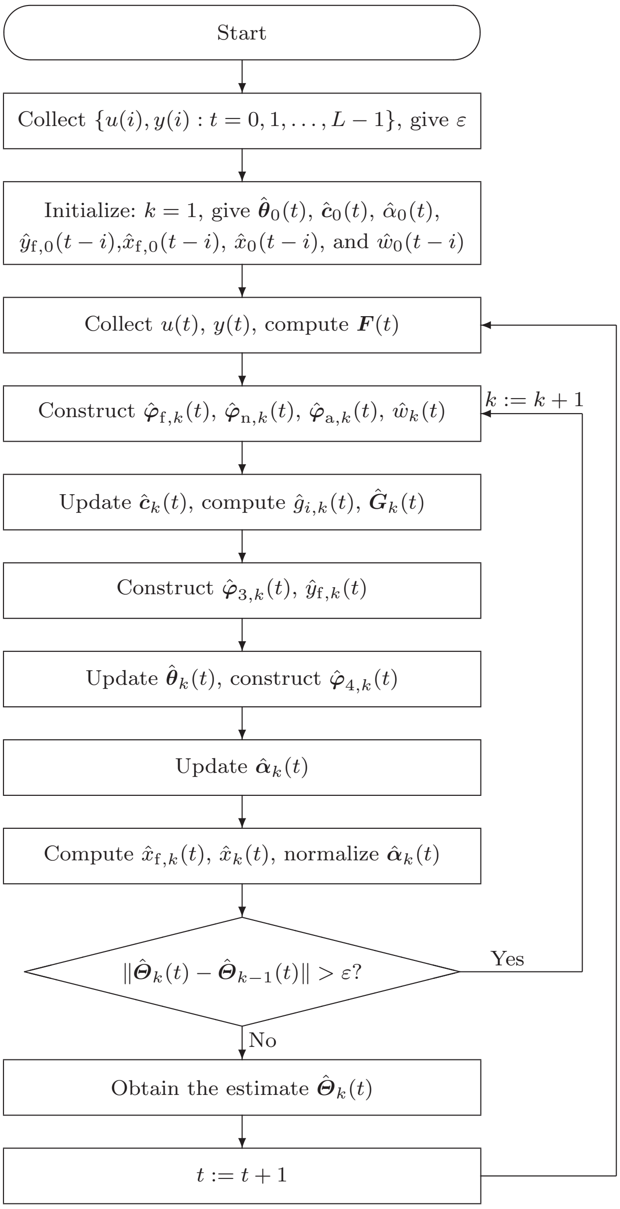

Start
Collect $\{u(i), y(i) : t = 0, 1, \dots, L-1\}$, give ε
Initialize: $k = 1$, give $\hat{\boldsymbol{\theta}}_0(t)$, $\hat{\boldsymbol{c}}_0(t)$, $\hat{\alpha}_0(t)$, $\hat{y}_{\mathrm{f},0}(t-i)$,$\hat{x}_{\mathrm{f},0}(t-i)$, $\hat{x}_0(t-i)$, and $\hat{w}_0(t-i)$
Collect $u(t)$, $y(t)$, compute $\boldsymbol{F}(t)$
$k := k+1$
Construct $\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t)$, $\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t)$, $\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t)$, $\hat{w}_k(t)$
Update $\hat{\boldsymbol{c}}_k(t)$, compute $\hat{g}_{i,k}(t)$, $\hat{\boldsymbol{G}}_k(t)$
Construct $\hat{\boldsymbol{\varphi}}_{3,k}(t)$, $\hat{y}_{\mathrm{f},k}(t)$
Update $\hat{\boldsymbol{\theta}}_k(t)$, construct $\hat{\boldsymbol{\varphi}}_{4,k}(t)$
Update $\hat{\boldsymbol{\alpha}}_k(t)$
Compute $\hat{x}_{\mathrm{f},k}(t)$, $\hat{x}_k(t)$, normalize $\hat{\boldsymbol{\alpha}}_k(t)$
$\|\hat{\boldsymbol{\Theta}}_k(t) - \hat{\boldsymbol{\Theta}}_{k-1}(t)\| > \varepsilon$?
Yes
No
Obtain the estimate $\hat{\boldsymbol{\Theta}}_k(t)$
$t := t+1$

Fig. 2 The flowchart of the F-D-LSI algorithm for computing $\hat{\boldsymbol{\theta}}_k(t)$, $\hat{\boldsymbol{\alpha}}_k(t)$ and $\hat{\boldsymbol{c}}_k(t)$

for estimating $\hat{\boldsymbol{\theta}}_k$, $\hat{\boldsymbol{\alpha}}_k$ and $\hat{\boldsymbol{c}}_k$ as follows:

$$\hat{\boldsymbol{\theta}}_k = \left[\sum_{j=1}^{L} \hat{\boldsymbol{\varphi}}_{3,k}(j)\hat{\boldsymbol{\varphi}}^{\mathrm{T}}_{3,k}(j)\right]^{-1} \sum_{j=1}^{L} \hat{\boldsymbol{\varphi}}_{3,k}(j)\hat{y}_{\mathrm{f},k}(j), \tag{58}$$

$$\hat{\boldsymbol{\varphi}}_{3,k}(t) = [\hat{\boldsymbol{\varphi}}^{\mathrm{T}}_{\mathrm{f},k}(t), \hat{\boldsymbol{\alpha}}^{\mathrm{T}}_{k-1}\hat{\boldsymbol{G}}^{\mathrm{T}}_k(t)]^{\mathrm{T}}, \quad t = 1, 2, \dots, L, \tag{59}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t) = [-\hat{x}_{\mathrm{f},k-1}(t-1), -\hat{x}_{\mathrm{f},k-1}(t-2), \dots, -\hat{x}_{\mathrm{f},k-1}(t-n_a)]^{\mathrm{T}}, \tag{60}$$

$$\hat{x}_{\mathrm{f},k}(t) = \hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t)\hat{\boldsymbol{a}}_k + \hat{\boldsymbol{b}}^{\mathrm{T}}_k\hat{\boldsymbol{G}}_k(t)\hat{\boldsymbol{\alpha}}_k, \tag{61}$$

$$\hat{\boldsymbol{\alpha}}_k = \left[\sum_{j=1}^{L} \hat{\boldsymbol{\varphi}}_{4,k}(j)\hat{\boldsymbol{\varphi}}^{\mathrm{T}}_{4,k}(j)\right]^{-1} \sum_{j=1}^{L} \hat{\boldsymbol{\varphi}}_{4,k}(j)[\hat{y}_{\mathrm{f},k}(j) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}_{\mathrm{f},k}(j)\hat{\boldsymbol{a}}_k], \tag{62}$$

$$\hat{\boldsymbol{\varphi}}_{4,k}(t) = \hat{\boldsymbol{G}}^{\mathrm{T}}_k(t)\hat{\boldsymbol{b}}_k, \tag{63}$$

$$\hat{\boldsymbol{G}}_k(t)=\left[\begin{array}{cccc} \hat{g}_{1,k}(t-1) & \hat{g}_{2,k}(t-1) & \ldots & \hat{g}_{m,k}(t-1) \\ \hat{g}_{1,k}(t-2) & \hat{g}_{2,k}(t-2) & \ldots & \hat{g}_{m,k}(t-2) \\ \vdots & \vdots & & \vdots \\ \hat{g}_{1,k}(t-n_b) & \hat{g}_{2,k}(t-n_b) & \ldots & \hat{g}_{m,k}(t-n_b) \end{array}\right], \tag{64}$$

$$\hat{g}_{j,k}(t)=f_j(u(t))+\hat{c}_{1,k}f_j(u(t-1))+\hat{c}_{2,k}f_j(u(t-2))+\ldots+\hat{c}_{n_c,k}f_j(u(t-n_c)), \tag{65}$$

$$\hat{y}_{\mathrm{f},k}(t)=y(t)+\hat{c}_{1,k}y(t-1)+\hat{c}_{2,k}y(t-2)+\ldots+\hat{c}_{n_c,k}y(t-n_c), \tag{66}$$

$$\hat{\boldsymbol{c}}_k=\left[\sum_{j=1}^{L}\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(j)\hat{\boldsymbol{\varphi}}^{\mathrm{T}}_{\mathrm{n},k}(j)\right]^{-1}\sum_{j=1}^{L}\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(j)\hat{w}_k(j), \tag{67}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t)=[-\hat{w}_{k-1}(t-1),-\hat{w}_{k-1}(t-2),\ldots,-\hat{w}_{k-1}(t-n_c)]^{\mathrm{T}}, \tag{68}$$

$$\hat{w}_k(t)=y(t)-\hat{\boldsymbol{\varphi}}^{\mathrm{T}}_{\mathrm{a},k}(t)\hat{\boldsymbol{a}}_{k-1}-\hat{\boldsymbol{b}}^{\mathrm{T}}_{k-1}\boldsymbol{F}(t)\hat{\boldsymbol{\alpha}}_{k-1}, \tag{69}$$

$$\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t)=[-\hat{x}_{k-1}(t-1),-\hat{x}_{k-1}(t-2),\ldots,-\hat{x}_{k-1}(t-n_a)]^{\mathrm{T}}, \tag{70}$$

$$\hat{x}_k(t)=[\hat{\boldsymbol{\varphi}}^{\mathrm{T}}_{\mathrm{a},k}(t),\hat{\boldsymbol{\alpha}}^{\mathrm{T}}_k\boldsymbol{F}^{\mathrm{T}}(t)]\hat{\boldsymbol{\theta}}_k, \tag{71}$$

$$\boldsymbol{F}(t)=\left[\begin{array}{cccc} f_1(u(t-1)) & f_2(u(t-1)) & \ldots & f_m(u(t-1)) \\ f_1(u(t-2)) & f_2(u(t-2)) & \ldots & f_m(u(t-2)) \\ \vdots & \vdots & & \vdots \\ f_1(u(t-n_b)) & f_2(u(t-n_b)) & \ldots & f_m(u(t-n_b)) \end{array}\right], \tag{72}$$

$$\hat{\boldsymbol{a}}_k=\hat{\boldsymbol{\theta}}_k(1:n_a), \tag{73}$$

$$\hat{\boldsymbol{b}}_k=\hat{\boldsymbol{\theta}}_k(1+n_a:n_a+n_b), \tag{74}$$

$$\bar{\boldsymbol{\alpha}}_k=\mathrm{sgn}[\hat{\boldsymbol{\alpha}}_k(1)]\frac{\hat{\boldsymbol{\alpha}}_k}{\|\hat{\boldsymbol{\alpha}}_k\|},\quad \hat{\boldsymbol{\alpha}}_k=\bar{\boldsymbol{\alpha}}_k, \tag{75}$$

$$\hat{\boldsymbol{c}}_k=[\hat{c}_{1,k},\hat{c}_{2,k},\ldots,\hat{c}_{n_c,k}]^{\mathrm{T}}, \tag{76}$$

$$\hat{\boldsymbol{\Theta}}_k=[\hat{\boldsymbol{\theta}}^{\mathrm{T}}_k,\hat{\boldsymbol{\alpha}}^{\mathrm{T}}_k,\hat{\boldsymbol{c}}^{\mathrm{T}}_k]^{\mathrm{T}}. \tag{77}$$

The flowchart of computing the parameter estimate $\hat{\boldsymbol{\Theta}}_k$ in the F-D-LSI algorithm in (58)–(77) with finite measurement data is shown in Fig. 3.

The F-D-LSI algorithm can be used to identify input nonlinear systems (Hammerstein nonlinear systems). The typical example is the first-order water tank system in Fig. 4, where $u(t)$ is the valve opening, $\bar{u}(t)$ is the water inlet flow, and $y(t)$ is the liquid level, the transfer function of the linear dynamical block has the form of $\frac{b_1z^{-1}}{1+a_1z^{-1}}$. The nonlinearity of the valve can be approximately fitted by a polynomial or a linear combination of the known base functions and the disturbance is an autoregressive process $w(t):=\frac{1}{1+c_1z^{-1}}v(t)$, $v(t)$ is white noise. The diagram of the water tank setup is shown in Fig. 5. Thus, the proposed F-D-LSI algorithm can be applied to such a system.

6 Example

Consider a Hammerstein nonlinear simulation model as follow:

$$\begin{aligned} y(t)&=\frac{B(z)}{A(z)}\bar{u}(t)+\frac{1}{C(z)}v(t),\\ \bar{u}(t)&=\alpha_1u^2(t)+\alpha_2u^3(t)=0.80u^2(t)+0.60u^3(t),\\ A(z)&=1+a_1z^{-1}+a_2z^{-2}=1+0.38z^{-1}+0.42z^{-2},\\ B(z)&=b_1z^{-1}+b_2z^{-2}=0.75z^{-1}-0.33z^{-2},\\ C(z)&=1+c_1z^{-1}=1+0.85z^{-1},\\ \boldsymbol{\theta}&=[0.38,0.42,0.75,-0.33,0.80,0.60,0.85]^{\mathrm{T}}. \end{aligned}$$

In simulation, the input $\{u(t)\}$ is taken as a persistent excitation signal sequence with zero mean and unit variance, and $\{v(t)\}$ as a white noise sequence with zero mean and variance σ^2, the data

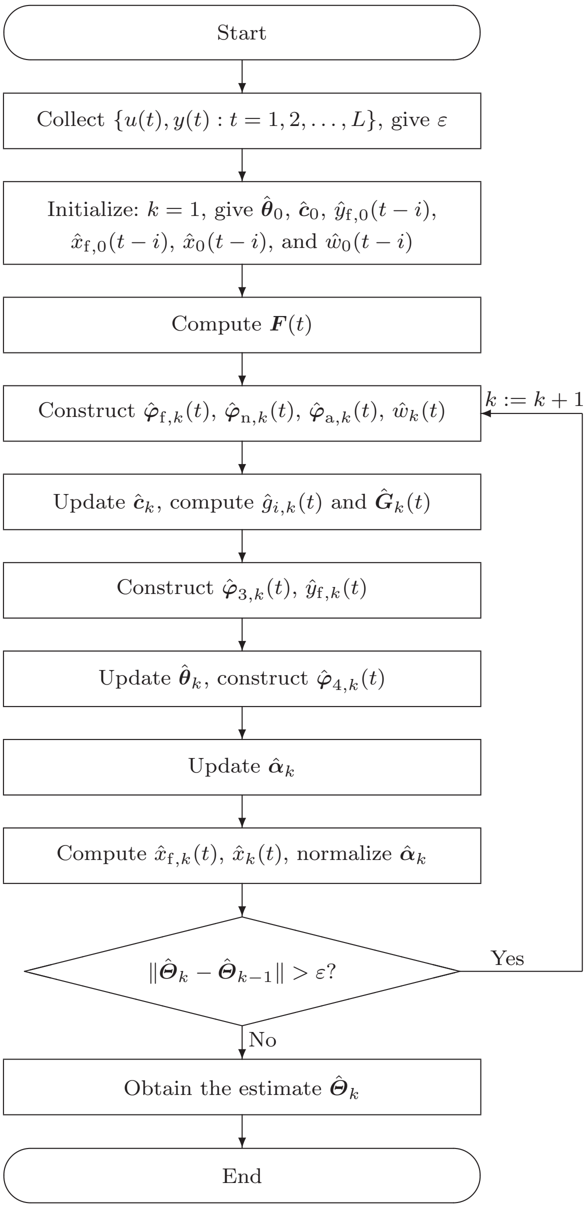

Start
Collect $\{u(t), y(t) : t = 1, 2, \ldots, L\}$, give ε
Initialize: $k = 1$, give $\hat{\boldsymbol{\theta}}_0$, $\hat{\boldsymbol{c}}_0$, $\hat{y}_{\mathrm{f},0}(t-i)$, $\hat{x}_{\mathrm{f},0}(t-i)$, $\hat{x}_0(t-i)$, and $\hat{w}_0(t-i)$
Compute $\boldsymbol{F}(t)$
Construct $\hat{\boldsymbol{\varphi}}_{\mathrm{f},k}(t)$, $\hat{\boldsymbol{\varphi}}_{\mathrm{n},k}(t)$, $\hat{\boldsymbol{\varphi}}_{\mathrm{a},k}(t)$, $\hat{w}_k(t)$
$k := k+1$
Update $\hat{\boldsymbol{c}}_k$, compute $\hat{g}_{i,k}(t)$ and $\hat{\boldsymbol{G}}_k(t)$
Construct $\hat{\boldsymbol{\varphi}}_{3,k}(t)$, $\hat{y}_{\mathrm{f},k}(t)$
Update $\hat{\boldsymbol{\theta}}_k$, construct $\hat{\boldsymbol{\varphi}}_{4,k}(t)$
Update $\hat{\boldsymbol{\alpha}}_k$
Compute $\hat{x}_{\mathrm{f},k}(t)$, $\hat{x}_k(t)$, normalize $\hat{\boldsymbol{\alpha}}_k$
$\|\hat{\boldsymbol{\Theta}}_k - \hat{\boldsymbol{\Theta}}_{k-1}\| > \varepsilon$?
Yes
No
Obtain the estimate $\hat{\boldsymbol{\Theta}}_k$
End

Fig. 3 The flowchart of the F-D-LSI algorithm with finite measurement data for computing $\hat{\boldsymbol{\Theta}}_k$

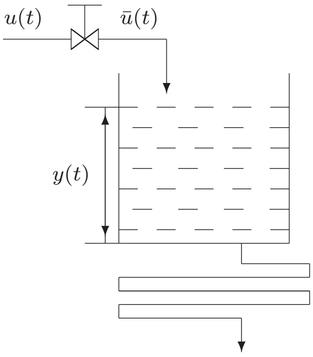


Fig. 4 An experimental setup of a water tank system

length $L = 1000$ and $L = 2000$, respectively. Applying the D-LSI algorithm in (15)–(26) and the F-D-LSI algorithm with finite measurement data in (58)–(77) to estimate the parameters of this

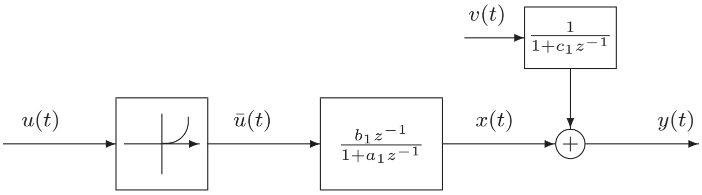


Fig. 5 The diagram of the water tank setup

system, the parameter estimates and their estimation errors $\delta := \|\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ with different data length L are shown in Tables 1–2, the F-D-LSI parameter estimation errors with different noise variances σ^2 are shown in Fig. 6, the parameter estimation errors of the two algorithms are plotted in Fig. 7.

When the noise variance $\sigma^2 = 1.00^2$, iteration $k = 15$, the D-LSI estimated model is given by

$$y(t) = \frac{0.76582z^{-1} - 0.31992z^{-2}}{1 + 0.38788z^{-1} + 0.37752z^{-2}}\bar{u}(t) + \frac{1}{1 + 0.85828z^{-1}}v(t),$$
$$\bar{u}(t) = 0.77549u^2(t) + 0.63137u^3(t),$$

the F-D-LSI estimated model is given by

$$y(t) = \frac{0.77354z^{-1} - 0.31140z^{-2}}{1 + 0.39827z^{-1} + 0.40385z^{-2}}\bar{u}(t) + \frac{1}{1 + 0.85839z^{-1}}v(t),$$
$$\bar{u}(t) = 0.79551u^2(t) + 0.60594u^3(t).$$

For model validation, we use a different data set ($L_e = 1000$ samples from $t = 2001$ to 3000) and the estimated models obtained by the D-LSI algorithm and the F-D-LSI algorithm. The predicted outputs and the true outputs are plotted in Fig. 8 from $t = 2001$ to 2100 and Fig. 9 from $t = 2001$ to 3000. Using the estimated outputs to compute the average output errors:

$$\delta_{e1} = \frac{1}{1000}\left[\sum_{j=2001}^{3000}[y(j) - \hat{y}_1(j)]^2\right]^{\frac{1}{2}} = 0.0550784,$$

$$\delta_{e2} = \frac{1}{1000}\left[\sum_{j=2001}^{3000}[y(j) - \hat{y}_2(j)]^2\right]^{\frac{1}{2}} = 0.0548344,$$

where $\hat{y}_1(t)$ is the predicted output given by the D-LSI model, $\hat{y}_2(t)$ is the predicted output given by the F-D-LSI model, and $y(t)$ is the true output.

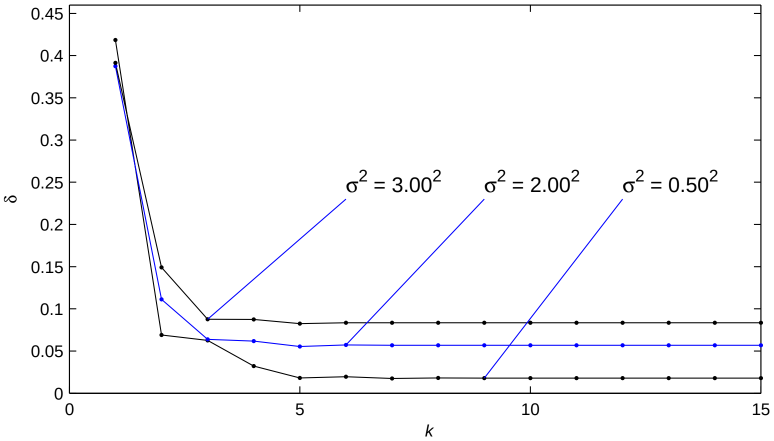


Fig. 6 The F-D-LSI estimation errors δ versus k ($L = 2000$)

From Figs. 6–9 and Tables 1–2, we can draw the following conclusions.

Table 1 The parameter estimates and errors versus iteration k ($\sigma^2 = 0.50^2$, $L = 1000$)

Algorithms	k	a_1	a_2	b_1	b_2	α_1	α_2	c_1	$\delta(\%)$
D-LSI	1	0.02094	-0.03204	0.74481	-0.55602	0.77692	0.62960	0.00093	63.84914
	2	0.43473	0.42385	0.77060	-0.33095	0.79820	0.60240	0.73608	7.77756
	3	0.33152	0.34580	0.77836	-0.38595	0.79127	0.61147	0.84967	6.64827
	4	0.33904	0.36807	0.77330	-0.37870	0.79340	0.60870	0.85005	5.22390
	5	0.35427	0.38317	0.77291	-0.36665	0.79227	0.61017	0.85142	3.86301
	10	0.35876	0.38411	0.76986	-0.36141	0.79072	0.61218	0.85202	3.51741
F-D-LSI	1	0.04618	-0.02019	0.75006	-0.57505	0.88337	0.66748	0.61513	39.88773
	2	0.35925	0.45558	0.76516	-0.37960	0.76723	0.61844	0.77757	6.37890
	3	0.30853	0.36912	0.75579	-0.38307	0.78741	0.62341	0.84159	6.45868
	4	0.37483	0.39413	0.75879	-0.34759	0.78024	0.62273	0.85002	2.70707
	5	0.37107	0.40813	0.75494	-0.33713	0.78356	0.62274	0.85211	2.00202
	10	0.37561	0.40488	0.75717	-0.33660	0.78353	0.62138	0.85277	1.99342
True values		0.38000	0.42000	0.75000	-0.33000	0.80000	0.60000	0.85000	

Table 2 The parameter estimates and errors versus iteration k ($\sigma^2 = 0.50^2$, $L = 2000$)

Algorithms	k	a_1	a_2	b_1	b_2	α_1	α_2	c_1	$\delta(\%)$
D-LSI	1	-0.01200	0.00614	0.72453	-0.55148	0.77936	0.62658	-0.00384	63.79498
	2	0.46302	0.45050	0.74061	-0.29910	0.79530	0.60622	0.75542	8.11391
	3	0.33566	0.35793	0.76113	-0.36301	0.78854	0.61498	0.85624	5.23040
	4	0.36287	0.39897	0.75618	-0.34691	0.79055	0.61239	0.85650	2.22579
	5	0.38124	0.40648	0.75569	-0.33352	0.78855	0.61497	0.85919	1.56941
	10	0.37432	0.40012	0.75440	-0.33702	0.78826	0.61534	0.85936	1.87768
F-D-LSI	1	-0.00063	-0.00294	0.73928	-0.57364	0.88889	0.66575	0.57716	41.60456
	2	0.37176	0.46262	0.75183	-0.36609	0.76951	0.61516	0.77109	6.24188
	3	0.31375	0.38596	0.74718	-0.37215	0.79236	0.61911	0.84758	5.34710
	4	0.37879	0.39422	0.75172	-0.33997	0.78415	0.61771	0.85527	2.23945
	5	0.38252	0.42147	0.74641	-0.32178	0.78701	0.61801	0.85822	1.54725
	10	0.38485	0.41473	0.74928	-0.32332	0.78753	0.61631	0.85903	1.48614
True values		0.38000	0.42000	0.75000	-0.33000	0.80000	0.60000	0.85000	

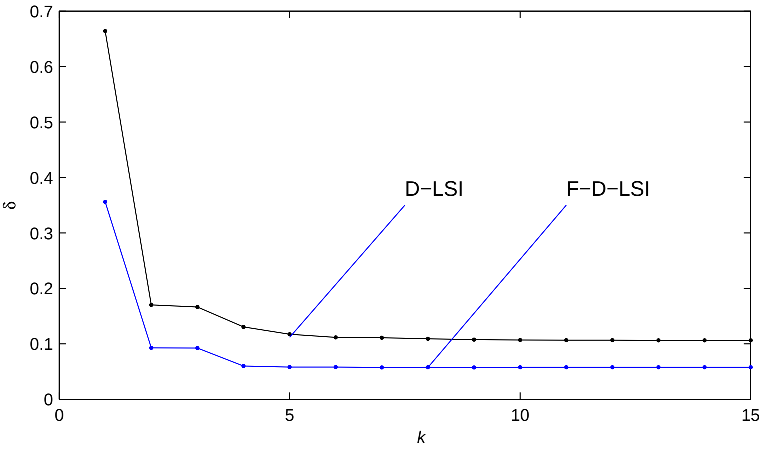
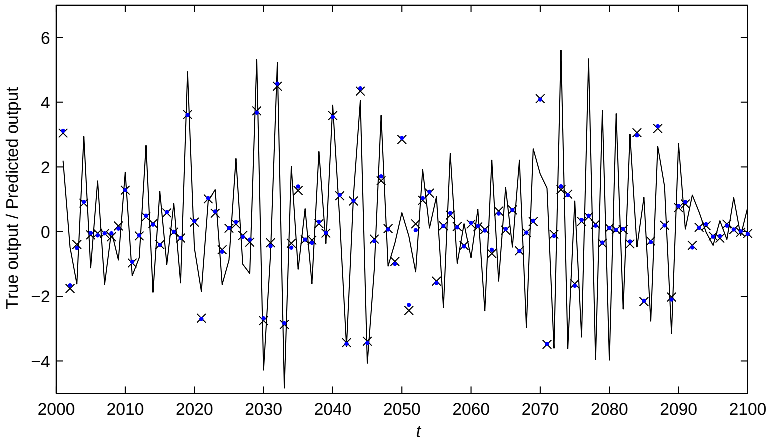


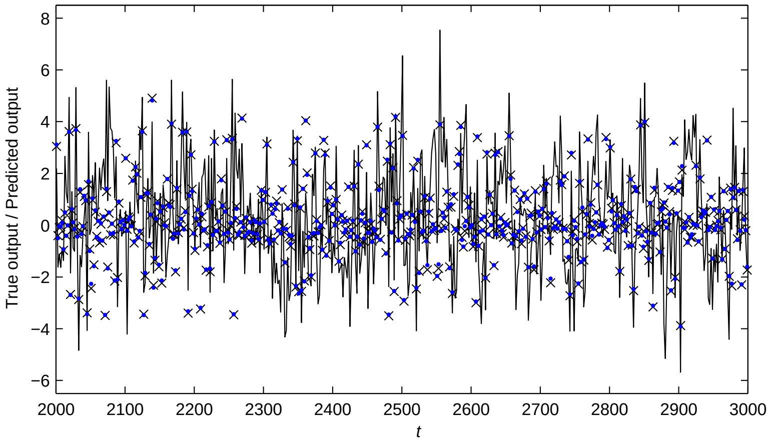
Fig. 7 The parameter estimation errors δ versus k ($\sigma^2 = 1.50^2$, $L = 1000$)

- The parameter estimation errors are becoming smaller (in general) as k increasing – see Fig. 6 and Fig. 7.
- Under the same data length, the parameter estimation errors become smaller as the noise variances decrease – see Fig. 6.
- Under the same noise variances and data lengths, the F-D-LSI algorithm can generate more accurate parameter estimates than the F-LSI algorithm – see Tables 1–2 and Fig. 7.
- The F-D-LSI algorithm can generate accurate parameter estimates after only several iterations – see Tables 1–2.



Solid line: $y(t)$; cross: $\hat{y}_1(t)$; dots: $\hat{y}_2(t)$

Fig. 8 The true output and predicted output from $t = 2001$ to 2100 ($\sigma^2 = 1.00^2$, $k = 15$)



Solid line: $y(t)$; cross: $\hat{y}_1(t)$; dots: $\hat{y}_2(t)$

Fig. 9 The true output and predicted output from $t = 2001$ to 3000 ($\sigma^2 = 1.00^2$, $k = 15$)

– The predicted outputs are very close to the true outputs, so the estimated model can capture the dynamics system well – see Figs. 8–9.

7 Conclusions

This paper presents a least squares iterative algorithm and a filtering based least squares iterative algorithm for IN-OEAR systems by using the model decomposition technique. Compared with the D-LSI algorithm, the F-D-LSI algorithm has higher estimation accuracy. The simulation test validates the effectiveness of the proposed algorithms. The proposed algorithms can be extended to study the parameter estimation problem for dual-rate sampled systems and non-uniformly sampled systems [37–39] and Wiener nonlinear systems [40].

Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 61273194), the PAPD of Jiangsu Higher Education Institutions and the 111 Project (B12018).

References

1. Matinfar, M., Saeidy, M., Gharahsuflu, B., Eslami, M.: Solutions of nonlinear chemistry problems by homotopy analysis. Comput. Math. Model. **25**(1), 103–114 (2014)
2. Kalafatis, A., Wang, L., Cluett, W.R.: Identification of time-varying pH processes using sinusoidal signals. Automatic **41**, 685–691 (2005)
3. Kitio Kwuimy, C. A., Litak, G., Nataraj, C.: Nonlinear analysis of energy harvesting systems with fractional order physical properties. Nonlinear Dyn. **80**(1–2), 491–501 (2015)
4. Huang, J., Shi ,Y., Huang, H.N., Li,Z.: l-2–l-infinity filtering for multirate nonlinear sampled-data systems using T-S fuzzy models. Digit. Signal Process. **23**(1), 418–426 (2013)
5. Ji, Y., Liu, X.M., Ding, F.: New criteria for the robust impulsive synchronization of uncertain chaotic delayed nonlinear systems, Nonlinear Dyn. **79**(1), 1–9 (2015)
6. Ji, Y., Liu, X.M.: Unified synchronization criteria for hybrid switching-impulsive dynamical networks, Circuits Syst. Signal Process. **34**(5), 1499–1517 (2015)
7. Li, H., Shi, Y.: Event-triggered robust model predictive control of continuous-time nonlinear systems. Automatica **50**(5), 1507–1513 (2014)
8. Guo, Z.K., Guan, X.P.: Nonlinear generalized predictive control based on online least squares support vector machines. Nonlinear Dyn. **79**(2), 1163–1168 (2015)
9. Vörös, J.: Iterative identification of nonlinear dynamic systems with input saturation and output backlash using three-block cascade models. J. Franklin Inst. **351**(12), 5455–5466 (2014)
10. Zhang, D.L., Tang, Y.G., Ma, J.H., Guan, X.P.: Identification of wiener model with discontinuous nonlinearities using differential evolution. Int.J. Control, Autom. Syst. **11**(3), 511–518 (2013)
11. Xu, L.: Application of the Newton iteration algorithm to the parameter estimation for dynamical systems. J. Comput. Appl. Math. **288**, 33–43 (2015)
12. Xu, L.: A proportional differential control method for a time-delay system using the Taylor expansion approximation. Appl. Math. Comput. **236**, 391–399 (2014)
13. Xu, L., Chen, L., Xiong, W.L.: Parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration. Nonlinear Dyn. **79**(3), 2155–2163 (2015)
14. Hagenblad, A., Ljung, L., Wills, A.: Maximum likelihood identification of Wiener models. Automatica **44**(11), 2697–2705 (2008)
15. Vörös, J.: Iterative identification of nonlinear dynamic systems with output backlash using three-block cascade models. Nonlinear Dyn. **79**(3), 2187–2195 (2015)
16. Hu, Y.B., Liu, B.L., Zhou, Q., Yang, C.: Recursive extended least squares parameter estimation for Wiener nonlinear systems with moving average noises. Circuits Syst. Signal Process. **33**(2), 655–664 (2014)
17. Paduart, J., Lauwers, L., Pintelon, R., Schoukens, J.: Identification of a Wiener–Hammerstein system using the polynomial nonlinear state space approach. Control Eng. Practice **20**(11), 1133–1139 (2012)
18. Sun, J.L., Liu, X.G.: A novel APSO-aided maximum likelihood identification method for Hammerstein systems. Nonlinear Dyn. **73**(1–2), 449–462 (2013)
19. Li, K., Peng, J.X., Bai, E.W.: A two-stage algorithm for identification of nonlinear dynamic systems, Automatica **42**(7), 1189–1197 (2006)
20. Mousazadeh, S., Karimi, M.: Estimating multivariate ARCH parameters by two-stage least-squares method. Signal Process. **89**(5), 921–932 (2009)
21. Zhang, W.G.: Decomposition based least squares iterative estimation algorithm for output error moving average systems. Eng. Comput. **31**(4), 709–725 (2014)
22. Bai, E.W., Liu, Y.: Least squares solutions of bilinear equations. Syst. Control Lett. **55**(6), 466–472 (2006)
23. Wang, D.Q. Ding, F.: Least squares based and gradient based iterative identification for Wiener nonlinear systems. Signal Process. **91**(5), 1182–1189 (2011)
24. Shi Y., Fang H.: Kalman filter based identification for systems with randomly missing measurements in a network environment. Int. J. Control **83**(3), 538–551 (2010)
25. Kohli, A.K., Amrita, R.: Numeric variable forgetting factor RLS algorithm for second-order volterra filtering. Circuits Syst. Signal Process. **32**(1), 223–232 (2013)
26. Prakash, J., Huang, B., Shah, S.L.: Recursive constrained state estimation using modified extended Kalman filter. Comput. Chem. Eng. **65**, 9–17 (2014)
27. Zhao, Z.G., Huang, B., Liu, F.: Parameter estimation in batch process using EM algorithm with particle filter. Comput. Chem. Eng. **57**, 159–172 (2013)
28. Ding, F., Wang, Y.J., Ding, J.: Recursive least squares parameter identification for systems with colored noise using the filtering technique and the auxiliary model, Digit. Signal Process. **37**, 100–108 (2015)
29. Wang C., Tang T.: Several gradient-based iterative estimation algorithms for a class of nonlinear systems using the filtering technique. Nonlinear Dyn. **77**(3), 769–780 (2014)
30. Wang, D.Q.: Least squares-based recursive and iterative estimation for output error moving average systems using data filtering. IET Control Theory Appl. **5**(14), 1648–1657 (2011)
31. Ding, F., Chen, T.: Identification of Hammerstein nonlinear ARMAX systems. Automatica **41**(9), 1479–1489 (2005)
32. Ding, F.: Decomposition based fast least squares algorithm for output error systems. Signal Process. **93**(5), 1235–1242 (2013)
33. Li, J.H.: Parameter estimation for Hammerstein CARARMA systems based on the Newton iteration. Appl. Math. Lett. **26**(1), 91–96 (2013)
34. Liu, Y., Bai, E.W.: Iterative identification of Hammerstein systems. Automatica **43**(2), 346–354 (2007)
35. Abrahamsson, R., Kay, S.M., Stoica, P.: Estimation of the parameters of a bilinear model with applications to submarine detection and system identification. Digit. Signal Process. **17**(4), 756–773 (2007)

36. Y.N. Cao, Z.Q. Liu, Signal frequency and parameter estimation for power systems using the hierarchical dentification principle. Math. Comput. Model. **51**(5-6), 854–861 (2010)
37. Ding, J., Fan, C.X., Lin, J.X.: Auxiliary model based parameter estimation for dual-rate output error systems with colored noise. Appl. Math. Model. **37**(6), 4051–4058 (2013)
38. Ding, J., Lin, J.X.: Modified subspace identification for periodically non-uniformly sampled systems by using the lifting technique, Circuits Syst. Signal Process. **33**(5), 1439–1449 (2014)
39. Wang, D.Q., Liu, H.B., Ding, F.: Highly efficient identification methods for dual-rate Hammerstein systems. IEEE Trans. Control Syst. Technol. (2015), doi: 10.1109/TCST.2014.2387216
40. Ding, F., Ma, J.X., Xiao, Y.S.: Newton iterative identification for a class of output nonlinear systems with moving average noises. Nonlinear Dyn. **74**(1-2), 21–30 (2013)