Applied Mathematical Modelling 000 (2015) 1–13

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

The filtering based maximum likelihood recursive least squares estimation for multiple-input single-output systems☆

Feiyan Chen, Feng Ding*

Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, PR China

ARTICLE INFO

Article history:
Received 29 September 2014
Revised 24 June 2015
Accepted 13 August 2015
Available online xxx

Keywords:
System identification
Least squares
Maximum likelihood
Data filtering
Multiple-input systems

ABSTRACT

In this paper, we use a noise transfer function to filter the input–output data and propose a new recursive algorithm for multiple-input single-output systems under the maximum likelihood principle. The main contributions of this paper are to derive a filtering based maximum likelihood recursive least squares (F-ML-RLS) algorithm for reducing computational burden and to present two recursive least squares algorithms to show the effectiveness of the F-ML-RLS algorithm. In the end, an illustrative simulation example is provided to test the proposed algorithms and we show that the F-ML-RLS algorithm has a high computational efficiency with smaller sizes of its covariance matrices and can produce more accurate parameter estimates.

1. Introduction

Recursive algorithms have wide applications in many areas such as computational mathematics, system theory, and matrix equations [1–4]. For example, Dehghan and Hajarian proposed an iterative algorithm for solving the generalized coupled Sylvester matrix equations [5]; Hashemi and Dehghan proposed the use of an interval Gaussian elimination to find an enclosure for the united solution set of the interval matrix equation [6]; Dehghani-Madiseh and Dehghan introduced the generalized solution sets to the interval generalized Sylvester matrix equation and developed some algebraic approaches for inner and outer estimations [7].

The study of modeling and identification of multivariable systems has been receiving much attention because most of the realistic physical processes are multivariable systems [8–10]. Recently, there are many estimation methods developed for multivariable systems. For example, Zhang presented a recursive least squares estimation algorithm for the multi-input single-output systems based on the bias compensation technique [11]; Chen and Ding derived a decomposition based maximum likelihood generalized extended least squares algorithm for multiple-input single-output nonlinear Box–Jenkins systems [12].

The least squares algorithms have wide applications in signal processing [13,14], data filtering [15–18], system control [19–22] and system identification [23–25]. For example, Ding et al. proposed a recursive least squares parameter identification algorithms for output-error autoregressive systems [26]; Wang et al. presented a hierarchical least squares algorithm and a key term separation based least squares algorithm for dual-rate Hammerstein systems [27]; Hajarian and Dehghan proposed the generalized centro-symmetric and least squares generalized centro-symmetric solutions for solving a linear matrix equation [28].

☆ This work was supported by the National Natural Science Foundation of China (No. 61273194) and the PAPD of Jiangsu Higher Education Institutions.
* Corresponding author.
E-mail addresses: fychen12@126.com (F. Chen), fding@jiangnan.edu.cn (F. Ding).

http://dx.doi.org/10.1016/j.apm.2015.08.015

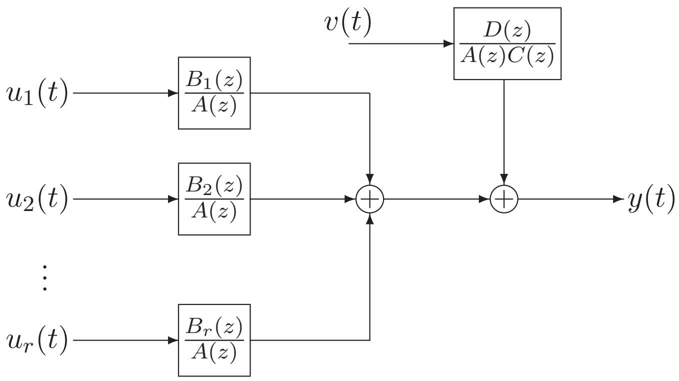

Fig. 1. The multiple-input single-output systems.

In the area of the maximum likelihood identification [29–31], Söderström et al. used the time domain maximum likelihood method and the sample maximum likelihood method to identify the errors-in-variables models under different assumptions and the results showed that these two methods have the same accuracy at any signal-to-noise ratios for output-error model identification [32]; Vanbeylen et al. constructed a Gaussian maximum likelihood estimator and proposed a blind maximum likelihood identification algorithm for discrete-time Hammerstein systems [33]; Chen et al. presented a maximum likelihood gradient-based iterative estimation algorithm for input nonlinear controlled autoregressive autoregressive moving average systems [34].

This paper studies the parameter estimation problem of a class of multiple-input single-output (MISO) systems with colored noise for given model representation with known structure. The identification method reported here is based on the maximum likelihood principle and thus differs from the hierarchical generalized least squares method in [35]. The proposed parameter estimation methods in this paper can be applied to study the modeling of other multivariable systems [36–38].

The outline of this paper is as follows. Section 2 derives a recursive generalized extended least squares algorithm for multiple-input single-output systems. Section 3 gives a filtering based recursive extended least squares algorithm. Section 4 derives a filtering based maximum likelihood recursive least squares identification algorithm and a recursive prediction error method. Section 5 provides numerical simulations to verify the effectiveness of the proposed algorithm. Finally, we offer some concluding remarks in Section 6.

2. The recursive generalized extended least squares algorithm

In this paper, we study the MISO system depicted in Fig. 1 and described by the following equation error model:

$$A(z)y(t) = \sum_{j=1}^{r} B_j(z)u_j(t) + \frac{D(z)}{C(z)}v(t), \tag{1}$$

where $y(t)$ is the system output, $u_j(t), j = 1, 2, \ldots, r$, are the system inputs, $v(t)$ is the uncorrelated stochastic noise with zero mean and variance σ^2, $A(z)$, $B_j(z)$, $C(z)$ and $D(z)$ are polynomials in the unit backward shift operator $z^{-1}[z^{-1}y(t) = y(t-1)]$, and

$$\begin{aligned}
A(z) &= 1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_{n_a}z^{-n_a},\\
B_j(z) &= b_{j1}z^{-1} + b_{j2}z^{-2} + \cdots + b_{jn_j}z^{-n_j},\\
C(z) &= 1 + c_1z^{-1} + c_2z^{-2} + \cdots + c_{n_c}z^{-n_c},\\
D(z) &= 1 + d_1z^{-1} + d_2z^{-2} + \cdots + d_{n_d}z^{-n_d}.
\end{aligned}$$

Assume that the order n_a, n_c, n_d and n_j, $j = 1, 2, \ldots, r$ are known and $y(t) = 0$, $u_j(t) = 0$ and $v(t) = 0$ as $t \leq 0$.
Define the inner variable

$$w(t) := \frac{D(z)}{C(z)}v(t), \tag{2}$$

which is an autoregressive moving average process. Let the superscript T denote the transpose and define the parameter vectors $\boldsymbol{\theta}$, $\boldsymbol{\theta}_{\rm s}$, $\boldsymbol{\theta}_{\rm n}$ and the information vectors $\boldsymbol{\varphi}(t)$, $\boldsymbol{\varphi}_{\rm s}(t)$, $\boldsymbol{\varphi}_{\rm n}(t)$ as

$$\begin{aligned}
\boldsymbol{\theta} &:= \begin{bmatrix}\boldsymbol{\theta}_{\rm s}\\ \boldsymbol{\theta}_{\rm n}\end{bmatrix} \in \mathbb{R}^n,\ n := n_a + \sum_{i=1}^{r} n_i + n_c + n_d,\\
\boldsymbol{\theta}_{\rm s} &:= [a_1, a_2, \ldots, a_{n_a}, b_{11}, b_{12}, \ldots, b_{1n_1}, b_{21}, b_{22}, \ldots, b_{2n_2}, \ldots, b_{r1}, b_{r2}, \ldots, b_{rn_r}]^{\rm T} \in \mathbb{R}^{n_a+n_1+n_2+\cdots+n_r},\\
\boldsymbol{\theta}_{\rm n} &:= [c_1, c_2, \ldots, c_{n_c}, d_1, d_2, \ldots, d_{n_d}]^{\rm T} \in \mathbb{R}^{n_c+n_d},
\end{aligned}$$

$$\boldsymbol{\varphi}(t) := \begin{bmatrix} \boldsymbol{\varphi}_{\rm s}(t) \\ \boldsymbol{\varphi}_{\rm n}(t) \end{bmatrix} \in \mathbb{R}^{n},$$

$$\boldsymbol{\varphi}_{\rm s}(t) := [-y(t-1), -y(t-2), \ldots, -y(t-n_a), u_1(t-1), u_1(t-2), \ldots, u_1(t-n_1), u_2(t-1), u_2(t-2), \ldots, u_2(t-n_2), \ldots, u_r(t-1), u_r(t-2), \ldots, u_r(t-n_r)]^{\rm T} \in \mathbb{R}^{n_a+n_1+n_2+\cdots+n_r},$$

$$\boldsymbol{\varphi}_{\rm n}(t) := [-w(t-1), -w(t-2), \ldots, -w(t-n_c), v(t-1), v(t-2), \ldots, v(t-n_d)]^{\rm T} \in \mathbb{R}^{n_c+n_d}.$$

Then Eq. (2) can be written as

$$w(t) = [1-C(z)]w(t) + D(z)v(t) = \boldsymbol{\varphi}_{\rm n}^{\rm T}(t)\boldsymbol{\theta}_{\rm n} + v(t). \tag{3}$$

Using (2) and (3), Eq. (1) can be written as

$$y(t) = \boldsymbol{\varphi}_{\rm s}^{\rm T}(t)\boldsymbol{\theta}_{\rm s} + w(t) \tag{4}$$

$$= \boldsymbol{\varphi}^{\rm T}(t)\boldsymbol{\theta} + v(t). \tag{5}$$

Let $\hat{\boldsymbol{\theta}}(t)$ be the estimate of $\boldsymbol{\theta}$ at time t. The following recursive generalized extended least squares (RGELS) algorithm can identify the parameter vector $\hat{\boldsymbol{\theta}}(t)$:

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{P}(t)\hat{\boldsymbol{\varphi}}(t)[y(t) - \hat{\boldsymbol{\varphi}}^{\rm T}(t)\hat{\boldsymbol{\theta}}(t-1)], \tag{6}$$

$$\boldsymbol{P}(t) = \boldsymbol{P}(t-1) - \frac{\boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)\hat{\boldsymbol{\varphi}}^{\rm T}(t)\boldsymbol{P}(t-1)}{1 + \hat{\boldsymbol{\varphi}}^{\rm T}(t)\boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)}, \tag{7}$$

$$\hat{\boldsymbol{\theta}}(t) = \begin{bmatrix} \hat{\boldsymbol{\theta}}_{\rm s}(t) \\ \hat{\boldsymbol{\theta}}_{\rm n}(t) \end{bmatrix}, \quad \hat{\boldsymbol{\varphi}}(t) = \begin{bmatrix} \boldsymbol{\varphi}_{\rm s}(t) \\ \hat{\boldsymbol{\varphi}}_{\rm n}(t) \end{bmatrix}, \tag{8}$$

$$\boldsymbol{\varphi}_{\rm s}(t) = [-y(t-1), -y(t-2), \ldots, -y(t-n_a), u_1(t-1), u_1(t-2), \ldots, u_1(t-n_1), u_2(t-1), u_2(t-2), \ldots, u_2(t-n_2), \ldots, u_r(t-1), u_r(t-2), \ldots, u_r(t-n_r)]^{\rm T}, \tag{9}$$

$$\hat{\boldsymbol{\varphi}}_{\rm n}(t) = [-\hat{w}(t-1), -\hat{w}(t-2), \ldots, -\hat{w}(t-n_c), \hat{v}(t-1), \hat{v}(t-2), \ldots, \hat{v}(t-n_d)]^{\rm T}, \tag{10}$$

$$\hat{w}(t) = y(t) - \boldsymbol{\varphi}_s^{\rm T}(t)\hat{\boldsymbol{\theta}}_{\rm s}(t), \tag{11}$$

$$\hat{v}(t) = y(t) - \hat{\boldsymbol{\varphi}}^{\rm T}(t)\hat{\boldsymbol{\theta}}(t), \tag{12}$$

$$\hat{\boldsymbol{\theta}}_{\rm s}(t) = [\hat{a}_1(t), \hat{a}_2(t), \ldots, \hat{a}_{n_a}(t), \hat{b}_{11}(t), \hat{b}_{12}(t), \ldots, \hat{b}_{1n_1}(t), \hat{b}_{21}(t), \hat{b}_{22}(t), \ldots, \hat{b}_{2n_2}(t), \ldots, \hat{b}_{r1}(t), \hat{b}_{r2}(t), \ldots, \hat{b}_{rn_r}(t)]^{\rm T}, \tag{13}$$

$$\hat{\boldsymbol{\theta}}_{\rm n}(t) = [\hat{c}_1(t), \hat{c}_2(t), \ldots, \hat{c}_{n_c}(t), \hat{d}_1(t), \hat{d}_2(t), \ldots, \hat{d}_{n_d}(t)]^{\rm T}. \tag{14}$$

To summarize, we list the steps involved in the RGELS algorithm to recursively compute the parameter estimation vector $\hat{\boldsymbol{\theta}}(t)$ as t increases:

1. Let $t=1$, set the initial values $\hat{\boldsymbol{\theta}}(0) = \mathbf{1}_n/p_0$, $\boldsymbol{P}(0) = p_0\boldsymbol{I}$, where p_0 is a large number (e.g., $p_0 = 10^6$).
2. Collect the input–output data $u_j(t)$ and $y(t)$, form $\boldsymbol{\varphi}_s(t)$ and $\hat{\boldsymbol{\varphi}}_n(t)$ using (9) and (10), form $\hat{\boldsymbol{\varphi}}(t)$ using (8).
3. Compute $\boldsymbol{P}(t)$ by (7).
4. Update the parameter estimate $\hat{\boldsymbol{\theta}}(t)$ using (6).
5. Compute $\hat{w}(t)$ and $\hat{v}(t)$ using (11) and (12).
6. Increase t by 1 and go to Step 2.

3. The filtering based recursive extended least squares algorithm

In order to reduce the amount of calculation of the RGELS algorithm, we use the data filtering technique to derive a filtering based recursive extended least squares (F-RELS) algorithm in the following.

Define

$$u_{1j}(t) := C(z)u_j(t), \quad y_1(t) := C(z)y(t),$$

$$\boldsymbol{\varphi}_1(t) := \begin{bmatrix} \boldsymbol{\varphi}_a(t) \\ \boldsymbol{\varphi}_b(t) \\ \boldsymbol{\varphi}_d(t) \end{bmatrix}, \quad \boldsymbol{\theta}_1 := \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{d} \end{bmatrix},$$

$$\begin{aligned}
\boldsymbol{\varphi}_a(t) &:= [-y_1(t-1), -y_1(t-2), \ \ldots\ , -y_1(t-n_a)]^{\mathrm{T}} \in \mathbb{R}^{n_a},\\
\boldsymbol{\varphi}_b(t) &:= [u_1(t-1), u_1(t-2), \ \ldots\ , u_1(t-n_1), u_2(t-1), u_2(t-2), \ldots, u_2(t-n_2), \ \ldots\ ,\\
&\qquad u_r(t-1), u_r(t-2), \ \ldots\ , u_r(t-n_r)]^{\mathrm{T}} \in \mathbb{R}^{n_1+n_2+\cdots+n_r},\\
\boldsymbol{\varphi}_d(t) &:= [v(t-1), v(t-2), \ \ldots\ , v(t-n_d)]^{\mathrm{T}} \in \mathbb{R}^{n_d},\\
\boldsymbol{\varphi}_c(t) &:= [-w(t-1), -w(t-2), \ \ldots\ , -w(t-n_c)]^{\mathrm{T}} \in \mathbb{R}^{n_c},\\
\boldsymbol{\varphi}_0(t) &:= [-y(t-1), -y(t-2), \ \ldots\ , -y(t-n_a), u_1(t-1), u_1(t-2), \ldots, u_1(t-n_1), u_2(t-1),\\
&\qquad u_2(t-2), \ \ldots\ , u_2(t-n_2), \ \ldots\ , u_r(t-1), u_r(t-2), \ldots, u_r(t-n_r)]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_1+n_2+\cdots+n_r},\\
\boldsymbol{a} &:= [a_1, a_2, \ \ldots\ , a_{n_a}]^{\mathrm{T}} \in \mathbb{R}^{n_a},\\
\boldsymbol{b} &:= [b_{11}, b_{12}, \ \ldots\ , b_{1n_1}, b_{21}, b_{22}, \ \ldots\ , b_{2n_2}, \ldots, b_{r1}, b_{r2}, \ \ldots\ , b_{rn_r}]^{\mathrm{T}} \in \mathbb{R}^{n_1+n_2+\cdots+n_r},\\
\boldsymbol{c} &:= [c_1, c_2, \ \ldots\ , c_{n_c}]^{\mathrm{T}} \in \mathbb{R}^{n_c},\\
\boldsymbol{d} &:= [d_1, d_2, \ \ldots\ , d_{n_d}]^{\mathrm{T}} \in \mathbb{R}^{n_d},\\
\boldsymbol{\theta}_0 &:= [a_1, a_2, \ \ldots\ , a_{n_a}, b_{11}, b_{12}, \ \ldots\ , b_{1n_1}, b_{21}, b_{22}, \ \ldots\ , b_{2n_2}, \ \ldots\ , b_{r1}, b_{r2}, \ \ldots\ , b_{rn_r}]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_1+n_2+\cdots+n_r}.
\end{aligned}$$

Multiplying both sides of (1) by $C(z)$ gives

$$A(z)C(z)y(t) = \sum_{j=1}^{r} B_j(z)C(z)u_j(t) + D(z)v(t).$$

That is

$$A(z)y_1(t) = \sum_{j=1}^{r} B_j(z)u_{1j}(t) + D(z)v(t),$$

Then we have the identification model

$$\begin{aligned}
y_1(t) &= [1-A(z)]y_1(t) + \sum_{j=1}^{r} B_j(z)u_{1j}(t) + [D(z)-1]v(t) + v(t)\\
&= \boldsymbol{\varphi}_1^{\mathrm{T}}(t)\boldsymbol{\theta}_1 + v(t). \qquad (15)
\end{aligned}$$

From (2), we have

$$\begin{aligned}
w(t) &= [1-C(z)]w(t) + [D(z)-1]v(t) + v(t)\\
&= \boldsymbol{\varphi}_c^{\mathrm{T}}(t)\boldsymbol{c} + \boldsymbol{\varphi}_d^{\mathrm{T}}(t)\boldsymbol{d} + v(t) \qquad (16)\\
&= y(t) - [1-A(z)]y(t) - \sum_{j=1}^{r} B_j(z)u_j(t)\\
&= y(t) - \boldsymbol{\varphi}_0^{\mathrm{T}}(t)\boldsymbol{\theta}_0,
\end{aligned}$$

For the identification models in (15) and (16), we can obtain a filtering based recursive extended least squares algorithm (F-RELS):

$$\hat{\boldsymbol{\theta}}_1(t) = \hat{\boldsymbol{\theta}}_1(t-1) + \boldsymbol{L}_1(t)[\hat{y}_1(t) - \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_1(t-1)], \qquad (17)$$

$$\boldsymbol{L}_1(t) = \frac{\boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_1(t)}{1 + \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_1(t)}, \qquad (18)$$

$$\boldsymbol{P}_1(t) = [\boldsymbol{I} - \boldsymbol{L}_1(t)\hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)]\boldsymbol{P}_1(t-1), \boldsymbol{P}_1(0) = p_0\boldsymbol{I}_{n_a+n_1+n_2+\cdots+n_r+n_d}, \qquad (19)$$

$$\begin{aligned}
\hat{\boldsymbol{\varphi}}_1(t) = [&-\hat{y}_1(t-1), -\hat{y}_1(t-2), \ \ldots\ , -\hat{y}_1(t-n_a), \hat{u}_{11}(t-1), \hat{u}_{11}(t-2), \ \ldots\ , \hat{u}_{11}(t-n_1),\\
&\hat{u}_{12}(t-1), \hat{u}_{12}(t-2), \ \ldots\ , \hat{u}_{12}(t-n_2), \ \ldots\ , \hat{u}_{1r}(t-1), \hat{u}_{1r}(t-2), \ \ldots\ , \hat{u}_{1r}(t-n_r),\\
&\hat{v}(t-1), \hat{v}(t-2), \ \ldots\ , \hat{v}(t-n_d)]^{\mathrm{T}}, \qquad (20)
\end{aligned}$$

$$\hat{y}_1(t) = y(t) + \hat{c}_1(t-1)y(t-1) + \hat{c}_2(t-1)y(t-2) + \ldots + \hat{c}_{n_c}(t-1)y(t-n_c), \qquad (21)$$

$$\hat{u}_{1j}(t) = u_j(t) + \hat{c}_1(t-1)u_j(t-1) + \hat{c}_2(t-1)u_j(t-2) + \ldots + \hat{c}_{n_c}(t-1)u_j(t-n_c), \qquad (22)$$

Please cite this article as: F. Chen, F. Ding, The filtering based maximum likelihood recursive least squares estimation for multiple-input single-output systems, Applied Mathematical Modelling (2015), http://dx.doi.org/10.1016/j.apm.2015.08.015

$$\hat{v}(t) = \hat{y}_1(t) - \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_1(t), \tag{23}$$

$$\hat{\boldsymbol{c}}(t) = \hat{\boldsymbol{c}}(t-1) + \boldsymbol{L}_c(t)[\hat{w}(t) - \hat{\boldsymbol{\varphi}}_c^{\mathrm{T}}(t)\hat{\boldsymbol{c}}(t-1) - \hat{\boldsymbol{\varphi}}_d^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t)], \tag{24}$$

$$\boldsymbol{L}_c(t) = \frac{\boldsymbol{P}_c(t-1)\hat{\boldsymbol{\varphi}}_c(t)}{1 + \hat{\boldsymbol{\varphi}}_c^{\mathrm{T}}(t)\boldsymbol{P}_c(t-1)\hat{\boldsymbol{\varphi}}_c(t)}, \tag{25}$$

$$\boldsymbol{P}_c(t) = [\boldsymbol{I} - \boldsymbol{L}_c(t)\hat{\boldsymbol{\varphi}}_c^{\mathrm{T}}(t)]\boldsymbol{P}_c(t-1), \boldsymbol{P}_c(0) = p_0\boldsymbol{I}_{n_c}, \tag{26}$$

$$\hat{\boldsymbol{\varphi}}_c(t) = [-\hat{w}(t-1), -\hat{w}(t-2), \ldots, -\hat{w}(t-n_c)]^{\mathrm{T}}, \tag{27}$$

$$\hat{w}(t) = y(t) - \boldsymbol{\varphi}_0^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_0(t), \tag{28}$$

$$\begin{aligned}\boldsymbol{\varphi}_0(t) = [&-y(t-1), -y(t-2), \ldots, -y(t-n_a), u_1(t-1), u_1(t-2), \ldots, u_1(t-n_1),\\ &u_2(t-1), u_2(t-2), \ldots, u_2(t-n_2), \ldots, u_r(t-1), u_r(t-2), \ldots, u_r(t-n_r)]^{\mathrm{T}}\end{aligned} \tag{29}$$

$$\hat{\boldsymbol{c}}(t) = [\hat{c}_1(t), \hat{c}_2(t), \ldots, \hat{c}_{n_c}(t)]^{\mathrm{T}}, \tag{30}$$

$$\begin{aligned}\hat{\boldsymbol{\theta}}_0(t) = [&\hat{a}_1(t), \hat{a}_2(t), \ldots, \hat{a}_{n_a}(t), \hat{b}_{11}(t), \hat{b}_{12}(t), \ldots, \hat{b}_{1n_1}(t), \hat{b}_{21}(t), \hat{b}_{22}(t), \ldots, \hat{b}_{2n_2}(t), \ldots,\\ &\hat{b}_{r1}(t), \hat{b}_{r2}(t), \ldots, \hat{b}_{rn_r}(t)]^{\mathrm{T}},\end{aligned} \tag{31}$$

$$\hat{\boldsymbol{\theta}}_1(t) = [\hat{\boldsymbol{\theta}}_0^{\mathrm{T}}(t), \hat{d}_1(t), \hat{d}_2(t), \ldots, \hat{d}_{n_d}(t)]^{\mathrm{T}}. \tag{32}$$

The steps involved in the F-RELS algorithm are listed in the following.

1. Let $t = 1$. Set the initial values $\boldsymbol{P}_1(0) = \boldsymbol{P}_c(0) = p_0\boldsymbol{I}$, $\hat{\boldsymbol{\theta}}_1(i) = \mathbf{1}_{n_a+n_b+n_1+n_2+\cdots+n_r}/p_0$, $\hat{\boldsymbol{c}}(i) = \mathbf{1}_{n_c}/p_0$, $\hat{y}_1(i) = 1/p_0$, $\hat{u}_{1j}(i) = 1/p_0$, $\hat{w}(i) = 1/p_0$ and $\hat{v}(i) = 1/p_0$ for $i \leq 0$, where p_0 is a large number (e.g., $p_0 = 10^6$).
2. Collect the input–output data $u_j(t)$ and $y(t)$, construct $\boldsymbol{\varphi}_0(t)$ using (29), $\hat{\boldsymbol{\varphi}}_1(t)$ using (20) and $\hat{\boldsymbol{\varphi}}_c(t)$ using (27).
3. Compute $\hat{y}_1(t)$ using (21) and $\hat{u}_{1j}(t)$ using (22).
4. Compute the gain vector $\boldsymbol{L}_1(t)$ using(18) and covariance matrix $\boldsymbol{P}_1(t)$ using (19), update the parameter estimate $\hat{\boldsymbol{\theta}}_1(t)$ using (17).
5. Compute $\hat{v}(t)$ using (23), $\hat{w}(t)$ using (28), the gain vector $\boldsymbol{L}_c(t)$ using (25) and the covariance matrix $\boldsymbol{P}_c(t)$ using (26).
6. Update the parameter estimate $\hat{\boldsymbol{c}}(t)$ using (24).
7. Increase t by 1 and go to Step 2.

4. The filtering based maximum likelihood recursive least squares algorithm

For the identification model in (15), the maximum likelihood estimation of $\boldsymbol{\theta}_1$ is obtained by maximizing the likelihood function, or the probability distribution function of the observation $u_N := \{u_1(1), u_1(2), \ldots, u_1(N)\}$ and $y_N := \{y_1(1), y_1(2), \ldots, y_1(N)\}$ conditioned on the parameter $\boldsymbol{\theta}_1$, i.e.

$$\hat{\boldsymbol{\theta}}_1 = \arg\max_{\boldsymbol{\theta}_1} L(y_N|u_{N-1}, \boldsymbol{\theta}_1).$$

Since the noise $v(t)$ is Gaussian white noise with zero mean and variance σ^2, the maximum likelihood function is given by

$$L(y_N|u_{N-1}, \boldsymbol{\theta}_1) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{N} v^2(t)\right) + \text{const.} \tag{33}$$

Taking the natural logarithm of both sides of (33) and maximizing the logarithm likelihood function give the following equivalent cost function

$$J(\boldsymbol{\theta}_1) := \frac{1}{N}\sum_{t=1}^{N} v^2(t) = \min,$$

$$v(t) = \frac{1}{D(z)}[A(z)y_1(t) - \sum_{j=1}^{r} B_j(z)u_{1j}(t)]. \tag{34}$$

Let $\hat{\boldsymbol{a}}(t), \hat{\boldsymbol{b}}(t), \hat{\boldsymbol{c}}(t), \hat{\boldsymbol{d}}(t)$, and $\hat{\boldsymbol{\theta}}_1(t)$ denote the estimates of $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$ and $\boldsymbol{\theta}_1$ at time t and

$$\hat{\boldsymbol{a}}(t) := [\hat{a}_1(t), \hat{a}_2(t), \ldots, \hat{a}_{n_a}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_a},$$

$$\begin{aligned}
\hat{\boldsymbol{b}}(t) &:= [\hat{b}_{11}(t), \hat{b}_{12}(t), \ldots, \hat{b}_{1n_1}(t), \hat{b}_{21}(t), \hat{b}_{22}(t), \ldots, \hat{b}_{2n_2}(t), \ldots, \\
&\quad \hat{b}_{r1}(t), \hat{b}_{r2}(t), \ldots, \hat{b}_{rn_r}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_1+n_2+\cdots+n_r}, \\
\hat{\boldsymbol{c}}(t) &:= [\hat{c}_1(t), \hat{c}_2(t), \ldots, \hat{c}_{n_c}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_c}, \\
\hat{\boldsymbol{d}}(t) &:= [\hat{d}_1(t), \hat{d}_2(t), \ldots, \hat{d}_{n_d}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_d}, \\
\hat{\boldsymbol{\theta}}_1(t) &:= \begin{bmatrix} \hat{\boldsymbol{a}}(t) \\ \hat{\boldsymbol{b}}(t) \\ \hat{\boldsymbol{d}}(t) \end{bmatrix} \in \mathbb{R}^{n_a+n_1+n_2+\cdots+n_r+n_d}.
\end{aligned}$$

Use the parameter estimates $\hat{c}(t)$ and $\hat{d}(t)$ to construct the estimates of $C(z)$ and $D(z)$ at time t:

$$\begin{aligned}
\hat{C}(t,z) &:= 1 + \hat{c}_1(t)z^{-1} + \hat{c}_2(t)z^{-2} + \cdots + \hat{c}_{n_c}(t)z^{-n_c}, \\
\hat{D}(t,z) &:= 1 + \hat{d}_1(t)z^{-1} + \hat{d}_2(t)z^{-2} + \cdots + \hat{d}_{n_d}(t)z^{-n_d}.
\end{aligned}$$

Computing the partial derivative of $v(t)$ in (34) with respect to a_j, $b_j(i)$, and d_j at the point $\hat{\boldsymbol{\theta}}_1(t-1)$ gives

$$\begin{aligned}
\left.\frac{\partial v(t)}{\partial a_j}\right|_{\hat{\boldsymbol{\theta}}_1(t-1)} &= \frac{\hat{C}(t-1,z)}{\hat{D}(t-1,z)} z^{-j} y(t) =: z^{-j}\hat{y}_f(t), \\
\left.\frac{\partial v(t)}{\partial b_j(i)}\right|_{\hat{\boldsymbol{\theta}}_1(t-1)} &= -\frac{\hat{C}(t-1,z)}{\hat{D}(t-1,z)} z^{-i} u_j(t) =: -z^{-i}\hat{u}_{jf}(t), \\
\left.\frac{\partial v(t)}{\partial d_j}\right|_{\hat{\boldsymbol{\theta}}_1(t-1)} &= -\frac{1}{\hat{D}(t-1,z)} z^{-j} \hat{v}(t) =: -z^{-j}\hat{v}_f(t),
\end{aligned}$$

where the filtered values $\hat{y}_f(t)$, $\hat{u}_{jf}(t)$and $\hat{v}_f(t)$ are defined as

$$\begin{aligned}
\hat{y}_f(t) &:= \frac{\hat{C}(t-1,z)}{\hat{D}(t-1,z)} y(t) \\
&= -\hat{d}_1(t-1)\hat{y}_f(t-1) - \hat{d}_2(t-1)\hat{y}_f(t-2) - \cdots - \hat{d}_{n_d}(t-1)\hat{y}_f(t-n_d) \\
&\quad + y(t) + \hat{c}_1(t-1)y(t-1) + \hat{c}_2(t-1)y(t-2) + \cdots + \hat{c}_{n_c}(t-1)y(t-n_c),
\end{aligned} \tag{35}$$

$$\begin{aligned}
\hat{u}_{jf}(t) &:= \frac{\hat{C}(t-1,z)}{\hat{D}(t-1,z)} u_j(t) \\
&= -\hat{d}_1(t-1)\hat{u}_{jf}(t-1) - \hat{d}_2(t-1)\hat{u}_{jf}(t-2) - \cdots - \hat{d}_{n_d}(t-1)\hat{u}_{jf}(t-n_d) \\
&\quad + u_j(t) + \hat{c}_1(t-1)u_j(t-1) + \hat{c}_2(t-1)u_j(t-2) + \cdots + \hat{c}_{n_c}(t-1)u_j(t-n_c),
\end{aligned} \tag{36}$$

$$\begin{aligned}
\hat{v}_f(t) &:= \frac{1}{\hat{D}(t-1,z)} \hat{v}(t) \\
&= \hat{v}(t) - \hat{d}_1(t-1)\hat{v}_f(t-1) - \hat{d}_2(t-1)\hat{v}_f(t-2) - \cdots - \hat{d}_{n_d}(t-1)\hat{v}_f(t-n_d).
\end{aligned} \tag{37}$$

Filtering $u_j(t)$ and $y(t)$ with $\hat{C}(t-1,z)$ to obtain the estimates of $u_{1j}(t)$ and $y_1(t)$:

$$\begin{aligned}
\hat{u}_{1j}(t) &= \hat{C}(t-1,z)u_j(t) = u_j(t) + \hat{c}_1(t-1)u_j(t-1) + \hat{c}_2(t-1)u_j(t-2) + \cdots + \hat{c}_{n_c}(t-1)u_j(t-n_c), \\
\hat{y}_1(t) &= \hat{C}(t-1,z)y(t) = y(t) + \hat{c}_1(t-1)y(t-1) + \hat{c}_2(t-1)y(t-2) + \cdots + \hat{c}_{n_c}(t-1)y(t-n_c).
\end{aligned}$$

Defined the filtered information vector

$$\begin{aligned}
\hat{\boldsymbol{\varphi}}_{1f}(t) &:= -\left.\frac{\partial v(t)}{\partial \boldsymbol{\theta}_1}\right|_{\hat{\boldsymbol{\theta}}_1(t-1)} \\
&= -\left[\frac{\partial v(t)}{\partial a_1}, \frac{\partial v(t)}{\partial a_2}, \ldots, \frac{\partial v(t)}{\partial a_{n_a}}, \frac{\partial v(t)}{\partial b_{11}}, \frac{\partial v(t)}{\partial b_{12}}, \ldots, \frac{\partial v(t)}{\partial b_{1n_1}}, \frac{\partial v(t)}{\partial b_{21}}, \frac{\partial v(t)}{\partial b_{22}}, \ldots, \frac{\partial v(t)}{\partial b_{2n_2}}, \ldots, \right. \\
&\quad \left. \frac{\partial v(t)}{\partial b_{r1}}, \frac{\partial v(t)}{\partial b_{r2}}, \ldots, \frac{\partial v(t)}{\partial b_{rn_r}} \frac{\partial v(t)}{\partial d_1}, \frac{\partial v(t)}{\partial d_2}, \ldots, \frac{\partial v(t)}{\partial d_{n_d}}\right]^{\mathrm{T}}_{\hat{\boldsymbol{\theta}}(t-1)} \\
&= [-\hat{y}_f(t-1), -\hat{y}_f(t-2), \ldots, -\hat{y}_f(t-n_a), \hat{u}_{1f}(t-1), \hat{u}_{1f}(t-2), \ldots, \hat{u}_{1f}(t-n_1), \\
&\quad \hat{u}_{2f}(t-1), \hat{u}_{2f}(t-2), \ldots, \hat{u}_{2f}(t-n_2), \ldots, \hat{u}_{rf}(t-1), \hat{u}_{rf}(t-2), \ldots, \hat{u}_{rf}(t-n_r), \\
&\quad \hat{v}_f(t-1), \hat{v}_f(t-2), \ldots, \hat{v}_f(t-n_d)]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_1+n_2+\cdots+n_r+n_d}.
\end{aligned}$$

Then we can obtain a filtering based maximum likelihood recursive least squares (F-ML-RLS) algorithm:

$$\hat{\boldsymbol{\theta}}_1(t) = \hat{\boldsymbol{\theta}}_1(t-1) + \boldsymbol{L}_1(t)\hat{v}(t), \tag{38}$$

$$\boldsymbol{L}_1(t) = \frac{\boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_{1f}(t)}{1+\hat{\boldsymbol{\varphi}}_{1f}^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_{1f}(t)}, \tag{39}$$

$$\boldsymbol{P}_1(t) = [\boldsymbol{I} - \boldsymbol{L}_1(t)\hat{\boldsymbol{\varphi}}_{1f}^{\mathrm{T}}(t)]\boldsymbol{P}_1(t-1), \tag{40}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}_1(t) = [&-\hat{y}_1(t-1), -\hat{y}_1(t-2), \ldots, -\hat{y}_1(t-n_a), \hat{u}_{11}(t-1), \hat{u}_{11}(t-2), \ldots, \hat{u}_{11}(t-n_1),\\ &\hat{u}_{12}(t-1), \hat{u}_{12}(t-2), \ldots, \hat{u}_{12}(t-n_2), \ldots, \hat{u}_{1r}(t-1), \hat{u}_{1r}(t-2), \ldots, \hat{u}_{1r}(t-n_r),\\ &\hat{v}(t-1), \hat{v}(t-2), \ldots, \hat{v}(t-n_d)]^{\mathrm{T}},\end{aligned} \tag{41}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}_{1f}(t) = [&-\hat{y}_f(t-1), -\hat{y}_f(t-2), \ldots, -\hat{y}_f(t-n_a), \hat{u}_{1f}(t-1), \hat{u}_{1f}(t-2), \ldots, \hat{u}_{1f}(t-n_1),\\ &\hat{u}_{2f}(t-1), \hat{u}_{2f}(t-2), \ldots, \hat{u}_{2f}(t-n_2), \ldots, \hat{u}_{rf}(t-1), \hat{u}_{rf}(t-2), \ldots, \hat{u}_{rf}(t-n_r),\\ &\hat{v}_f(t-1), \hat{v}_f(t-2), \ldots, \hat{v}_f(t-n_d)]^{\mathrm{T}},\end{aligned} \tag{42}$$

$$\hat{y}_1(t) = y(t) + \hat{c}_1(t-1)y(t-1) + \hat{c}_2(t-1)y(t-2) + \cdots + \hat{c}_{n_c}(t-1)y(t-n_c), \tag{43}$$

$$\hat{u}_{1j}(t) = u_j(t) + \hat{c}_1(t-1)u_j(t-1) + \hat{c}_2(t-1)u_j(t-2) + \cdots + \hat{c}_{n_c}(t-1)u_j(t-n_c), \tag{44}$$

$$\begin{aligned}\hat{y}_f(t) = &-\hat{d}_1(t-1)\hat{y}_f(t-1) - \hat{d}_2(t-1)\hat{y}_f(t-2) - \cdots - \hat{d}_{n_d}(t-1)\hat{y}_f(t-n_d)\\ &+ y(t) + \hat{c}_1(t-1)y(t-1) + \hat{c}_2(t-1)y(t-2) + \cdots + \hat{c}_{n_c}(t-1)y(t-n_c),\end{aligned} \tag{45}$$

$$\begin{aligned}\hat{u}_{jf}(t) = &-\hat{d}_1(t-1)\hat{u}_{jf}(t-1) - \hat{d}_2(t-1)\hat{u}_{jf}(t-2) - \cdots - \hat{d}_{n_d}(t-1)\hat{u}_{jf}(t-n_d)\\ &+ u_j(t) + \hat{c}_1(t-1)u_j(t-1) + \hat{c}_2(t-1)u_j(t-2) + \cdots + \hat{c}_{n_c}(t-1)u_j(t-n_c),\end{aligned} \tag{46}$$

$$\hat{v}_f(t) = \hat{v}(t) - \hat{d}_1(t-1)\hat{v}_f(t-1) - \hat{d}_2(t-1)\hat{v}_f(t-2) - \cdots - \hat{d}_{n_d}(t-1)\hat{v}_f(t-n_d), \tag{47}$$

$$\hat{v}(t) = \hat{y}_1(t) - \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_1(t-1), \tag{48}$$

$$\hat{\boldsymbol{c}}(t) = \hat{\boldsymbol{c}}(t-1) + \boldsymbol{L}_c(t)[\hat{w}(t) - \hat{\boldsymbol{\varphi}}_c^{\mathrm{T}}(t)\hat{\boldsymbol{c}}(t-1) - \hat{\boldsymbol{\varphi}}_d^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t)], \tag{49}$$

$$\boldsymbol{L}_c(t) = \frac{\boldsymbol{P}_c(t-1)\hat{\boldsymbol{\varphi}}_c(t)}{1+\hat{\boldsymbol{\varphi}}_c^{\mathrm{T}}(t)\boldsymbol{P}_c(t-1)\hat{\boldsymbol{\varphi}}_c(t)}, \tag{50}$$

$$\boldsymbol{P}_c(t) = [\boldsymbol{I} - \boldsymbol{L}_c(t)\hat{\boldsymbol{\varphi}}_c^{\mathrm{T}}(t)]\boldsymbol{P}_c(t-1), \tag{51}$$

$$\hat{\boldsymbol{\varphi}}_c(t) = [-\hat{w}(t-1), -\hat{w}(t-2), \ldots, -\hat{w}(t-n_c)]^{\mathrm{T}}, \tag{52}$$

$$\hat{w}(t) = y(t) - \boldsymbol{\varphi}_0^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_0(t), \tag{53}$$

$$\begin{aligned}\boldsymbol{\varphi}_0(t) = [&-y(t-1), -y(t-2), \ldots, -y(t-n_a), u_1(t-1), u_1(t-2), \ldots, u_1(t-n_1),\\ &u_2(t-1), u_2(t-2), \ldots, u_2(t-n_2), \ldots, u_r(t-1), u_r(t-2), \ldots, u_r(t-n_r)]^{\mathrm{T}}.\end{aligned} \tag{54}$$

To initialize the F-ML-RLS algorithm, we take

$$\hat{\boldsymbol{\theta}}_1(i) = \mathbf{1}_{n_a+n_b+n_1+n_2+\cdots+n_r}/p_0, \quad \hat{\boldsymbol{c}}(i) = \mathbf{1}_{n_c}/p_0, \ i \leqslant 0, \tag{55}$$

$$\boldsymbol{P}_1(0) = p_0\boldsymbol{I}_{n_a+n_b+n_1+n_2+\cdots+n_r}, \quad \boldsymbol{P}_c(0) = p_0\boldsymbol{I}_{n_c}, \ p_0 = 10^6. \tag{56}$$

The flowchart of computing the estimates $\hat{\boldsymbol{\theta}}_1(t)$ and $\hat{\boldsymbol{c}}(t)$ is shown in Fig. 2, the steps involved in the F-ML-RLS algorithm are listed in the following.

1. Let $t = 1$. Set the initial values of the parameter estimation vectors and covariance matrices using (55) and (56), and $\hat{y}_1(i) = 1/p_0, \hat{u}_{1j}(i) = 1/p_0, \hat{y}_f(i) = 1/p_0, \hat{u}_{jf}(i) = 1/p_0, \hat{v}_f(i) = 1/p_0, \hat{w}(i) = 1/p_0$ and $\hat{v}(i) = 1/p_0$ for $i \leqslant 0$.
2. Collect the input-output data $u_j(t)$ and $y(t)$, compute $\hat{y}_1(t)$ using (43), $\hat{u}_{1j}(t)$ using (44) and $\hat{v}(t)$ using(48).
3. Compute $\hat{y}_f(t)$, $\hat{u}_f(jf)$ and $\hat{v}_f(t)$ using (45), (46) and (47), respectively.

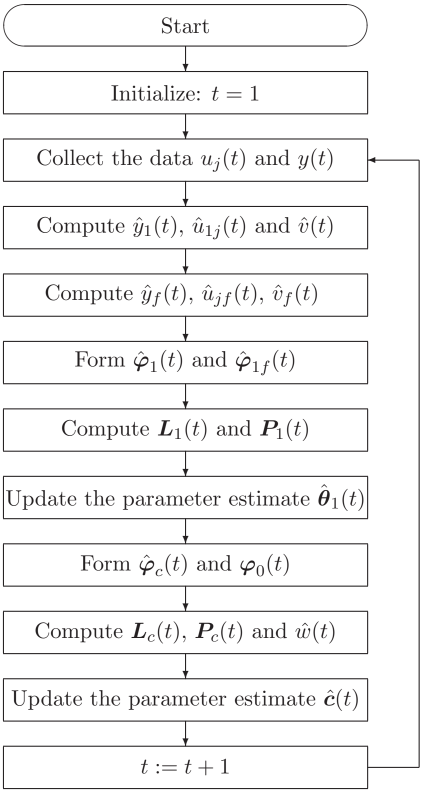

Fig. 2. The flowchart of computing the parameter estimate $\hat{\boldsymbol{\theta}}_1(t)$ and $\hat{\boldsymbol{c}}(t)$.

Table 1
The computational efficiency of the RGELS, F-RELS and F-ML-RLS algorithms.

Algorithms	RGELS	F-RELS	F-ML-RLS
Number of multiplication	$2n^2 + 4n + n_a + n_b$	$2m_1^2 + 2n_c^2 + rn_c + 5n$	$2m_1^2 + 2n_c^2 + 5n + rm_2 + m_3$
Number of addition	$2n^2 + 2n + n_a + n_b$	$2m_1^2 + 2n_c^2 + rn_c + 3n$	$2m_1^2 + 2n_c^2 + rm_2 + 3n + m_3$
Total flops	$N_1 := 4n^2 + 6n + 2(n_a + n_b)$	$N_2 := 4m_1^2 + 4n_c^2 + 2rn_c + 8n$	$N_3 := 4m_1^2 + 4n_c^2 + 2rm_2 + 8n + 2m_3$

4. Form $\hat{\boldsymbol{\varphi}}_1(t)$ using (41) and $\hat{\boldsymbol{\varphi}}_{1f}(t)$ using (42).
5. Compute gain vector $\boldsymbol{L}_1(t)$ and covariance matrix $\boldsymbol{P}_1(t)$ using (39) and (40), respectively.
6. Update the parameter estimate $\hat{\boldsymbol{\theta}}_1(t)$ using (38).
7. Form $\hat{\boldsymbol{\varphi}}_c(t)$ and $\boldsymbol{\varphi}_0(t)$ using (52) and (54), respectively. Compute $\boldsymbol{L}_c(t)$ using (50), the matrix $\boldsymbol{P}_c(t)$ using (51), and $\hat{w}(t)$ using (53).
8. Update the parameter estimate $\hat{\boldsymbol{c}}(t)$ using (49).
9. Increase t by 1 and go to Step 2.

We list the numbers of multiplications and additions of the F-ML-RLS, the F-RELS and the RGELS algorithms at each step to show the advantage of the proposed F-ML-RLS algorithm in Table 1, where the numbers in the brackets are the computation loads at each step with $n_b = n_1 + n_2$, $m_1 = n_a + n_b + n_d$, $m_2 = 2n_c + n_d$, $m_3 = n_c + 2n_d$ and $n = n_a + n_b + n_c + n_d$, we have $N_3 < N_1$, this shows that the F-ML-RLS algorithm has a higher computational efficiency than that of the RGELS algorithm.

Here, we give the recursive prediction error method (RPEM) for comparison [39,40]. Let the gradient vector $\boldsymbol{\varphi}_f(t) := \frac{\partial \hat{y}(t|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ with $\hat{y}(t|\boldsymbol{\theta})$ the prediction of the output regarding to $\boldsymbol{\theta}$. The prediction of the output of the system can be written as

$$\hat{y}(t|\boldsymbol{\theta}) := y(t) - \frac{C(z)}{D(z)}\left[A(z)y(t) - \sum_{j=1}^{r} B_j(z)u_j(t)\right],$$

then, the RPEM algorithm can be given by

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{P}(t)\hat{\boldsymbol{\varphi}}_f(t)e(t), \tag{57}$$

$$\boldsymbol{P}(t) = \rho(t)\boldsymbol{P}(t-1) - \frac{\rho^2(t)\boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}_f(t)\hat{\boldsymbol{\varphi}}_f^{\mathrm{T}}(t)\boldsymbol{P}(t-1)}{1+\rho(t)\hat{\boldsymbol{\varphi}}_f^{\mathrm{T}}(t)\boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}_f(t)},\ \boldsymbol{P}(0) = p_0\boldsymbol{I}, \tag{58}$$

$$e(t) = y(t) - \hat{y}(t) = y(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1), \tag{59}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}(t) = [&-y(t-1), -y(t-2), \dots, -y(t-n_a), u_1(t-1), u_1(t-2), \dots, u_1(t-n_1),\\ &u_2(t-1), u_2(t-2), \dots, u_2(t-n_2), \dots, u_r(t-1), u_r(t-2), \dots, u_r(t-n_r),\\ &-\hat{w}(t-1), -\hat{w}(t-2), \dots, -\hat{w}(t-n_c), \hat{v}(t-1), \hat{v}(t-2), \dots, \hat{v}(t-n_d)]^{\mathrm{T}},\end{aligned} \tag{60}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}_f(t) = [&-\hat{y}_f(t-1), -\hat{y}_f(t-2), \dots, -\hat{y}_f(t-n_a), \hat{u}_{1f}(t-1), \hat{u}_{1f}(t-2), \dots, \hat{u}_{1f}(t-n_1),\\ &\hat{u}_{2f}(t-1), \hat{u}_{2f}(t-2), \dots, \hat{u}_{2f}(t-n_2), \dots, \hat{u}_{rf}(t-1), \hat{u}_{rf}(t-2), \dots, \hat{u}_{rf}(t-n_r),\\ &-\hat{w}_f(t-1), -\hat{w}_f(t-2), \dots, \hat{w}_f(t-n_d), \hat{v}_f(t-1), \hat{v}_f(t-2), \dots, \hat{v}_f(t-n_d)]^{\mathrm{T}},\end{aligned} \tag{61}$$

$$\begin{aligned}\hat{y}_f(t) = &-\hat{d}_1(t-1)\hat{y}_f(t-1) - \hat{d}_2(t-1)\hat{y}_f(t-2) - \dots - \hat{d}_{n_d}(t-1)\hat{y}_f(t-n_d)\\ &+ y(t) + \hat{c}_1(t-1)y(t-1) + \hat{c}_2(t-1)y(t-2) + \dots + \hat{c}_{n_c}(t-1)y(t-n_c),\end{aligned} \tag{62}$$

$$\begin{aligned}\hat{u}_{jf}(t) = &-\hat{d}_1(t-1)\hat{u}_{jf}(t-1) - \hat{d}_2(t-1)\hat{u}_{jf}(t-2) - \dots - \hat{d}_{n_d}(t-1)\hat{u}_{jf}(t-n_d)\\ &+ u_j(t) + \hat{c}_1(t-1)u_j(t-1) + \hat{c}_2(t-1)u_j(t-2) + \dots + \hat{c}_{n_c}(t-1)u_j(t-n_c),\end{aligned} \tag{63}$$

$$\hat{w}(t) = y(t) - \boldsymbol{\varphi}_s^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_s(t), \tag{64}$$

$$\hat{w}_f(t) = \hat{w}(t) - \hat{d}_1(t-1)\hat{w}_f(t-1) - \hat{d}_2(t-1)\hat{w}_f(t-2) - \dots - \hat{d}_{n_d}(t-1)\hat{w}_f(t-n_d), \tag{65}$$

$$\hat{v}(t) = y(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t), \tag{66}$$

$$\hat{v}_f(t) = \hat{v}(t) - \hat{d}_1(t-1)\hat{v}_f(t-1) - \hat{d}_2(t-1)\hat{v}_f(t-2) - \dots - \hat{d}_{n_d}(t-1)\hat{v}_f(t-n_d), \tag{67}$$

$$\begin{aligned}\boldsymbol{\varphi}_s(t) = [&-y(t-1), -y(t-2), \dots, -y(t-n_a), u_1(t-1), u_1(t-2), \dots, u_1(t-n_1), u_2(t-1), u_2(t-2), \dots,\\ &u_2(t-n_2), \dots, u_r(t-1), u_r(t-2), \dots, u_r(t-n_r)]^{\mathrm{T}},\end{aligned} \tag{68}$$

$$\begin{aligned}\hat{\boldsymbol{\theta}}_s(t) = [&\hat{a}_1(t), \hat{a}_2(t), \dots, \hat{a}_{n_a}(t), \hat{b}_{11}(t), \hat{b}_{12}(t), \dots, \hat{b}_{1n_1}(t), \hat{b}_{21}(t), \hat{b}_{22}(t), \dots, \hat{b}_{2n_2}(t), \dots,\\ &\hat{b}_{r1}(t), \hat{b}_{r2}(t), \dots, \hat{b}_{rn_r}(t)]^{\mathrm{T}},\end{aligned} \tag{69}$$

$$\hat{\boldsymbol{\theta}}(t) = [\hat{\boldsymbol{\theta}}_s^{\mathrm{T}}(t), \hat{c}_1(t), \hat{c}_2(t), \dots, \hat{c}_{n_c}(t), \hat{d}_1(t), \hat{d}_2(t), \dots, \hat{d}_{n_d}(t)]^{\mathrm{T}}. \tag{70}$$

where $\rho(t)$ is a convergence factor, $e(t)$ is the prediction error or the system innovation.

5. Simulation study

This section provides an example to show the effectiveness of the proposed F-ML-RLS algorithm for the MISO systems, compared with the RGELS, F-RELS and RPEM algorithms.

Consider the following MISO system:

$$\begin{aligned}A(z)y(t) &= B_1(z)u_1(t) + B_2(z)u_2(t) + \frac{D(z)}{C(z)}v(t),\\ A(z) &= 1 + a_1z^{-1} + a_2z^{-2} = 1 - 0.35z^{-1} - 0.30z^{-2},\\ B_1(z) &= b_{11}z^{-1} + b_{12}z^{-2} = 0.15z^{-1} + 0.35z^{-2},\\ B_2(z) &= b_{21}z^{-1} + b_{22}z^{-2} = 0.18z^{-1} + 0.32z^{-2},\\ C(z) &= 1 + c_1z^{-1} = 1 - 0.05z^{-1},\\ D(z) &= 1 + d_1z^{-1} = 1 + 0.25z^{-1},\\ c_1 &= -0.05,\\ \boldsymbol{\theta}_1 &= [a_1, a_2, b_{11}, b_{12}, b_{21}, b_{22}, d_1]^{\mathrm{T}} = [-0.35, -0.30, 0.15, 0.35, 0.18, 0.32, 0.25]^{\mathrm{T}},\\ \boldsymbol{\theta} &= [a_1, a_2, b_{11}, b_{12}, b_{21}, b_{22}, d_1, c_1]^{\mathrm{T}} = [-0.35, -0.30, 0.15, 0.35, 0.18, 0.32, 0.25, -0.05]^{\mathrm{T}}.\end{aligned}$$

Table 2
The RGELS estimates and errors.

t	100	200	500	1000	2000	3000	True values
a_1	−0.04303	0.01583	−0.08066	−0.13681	−0.13093	−0.13587	−0.35000
a_2	−0.23553	−0.38357	−0.36773	−0.32368	−0.30527	−0.30672	−0.30000
b_{11}	0.15236	0.10849	0.13779	0.15594	0.14982	0.15050	0.15000
b_{12}	0.34453	0.33939	0.36705	0.35932	0.36149	0.35824	0.35000
b_{21}	0.20532	0.15405	0.13795	0.16657	0.16768	0.17548	0.18000
b_{22}	0.37649	0.37196	0.33144	0.30310	0.30454	0.29696	0.32000
d_1	−0.42584	−0.32982	−0.24739	−0.22274	−0.23344	−0.22634	0.25000
c_1	0.15933	0.40409	0.39938	0.34615	0.34494	0.34139	−0.05000
δ (%)	67.20848	66.71857	50.25448	39.20997	40.45503	39.27441	

Table 3
The F-RELS estimates and errors.

t	100	200	500	1000	2000	3000	True values
a_1	−0.16219	−0.22169	−0.26825	−0.30653	−0.32300	−0.33304	−0.35000
a_2	−0.28002	−0.36420	−0.36448	−0.33728	−0.31878	−0.31580	−0.30000
b_{11}	0.04596	0.04159	0.10276	0.13319	0.13874	0.14305	0.15000
b_{12}	0.16189	0.23997	0.31122	0.33098	0.34749	0.34807	0.35000
b_{21}	0.10850	0.14582	0.13713	0.16931	0.16736	0.17544	0.18000
b_{22}	0.33885	0.36897	0.32043	0.30550	0.30694	0.29854	0.32000
d_1	0.56975	0.47545	0.42954	0.38417	0.37084	0.35511	0.25000
c_1	0.09748	0.04650	−0.00185	−0.01093	−0.01158	−0.01258	−0.05000
δ (%)	61.58094	44.05243	30.24150	20.65547	17.78283	15.57521	

Table 4
The F-ML-RLS estimates and errors.

t	100	200	500	1000	2000	3000	True values
a_1	−0.33586	−0.35962	−0.31093	−0.36020	−0.33098	−0.34851	−0.35000
a_2	−0.35877	−0.30436	−0.32913	−0.29468	−0.31423	−0.30401	−0.30000
b_{11}	0.16321	0.11805	0.15041	0.14276	0.14326	0.13922	0.15000
b_{12}	0.34226	0.33218	0.34452	0.33542	0.34057	0.34086	0.35000
b_{21}	0.12629	0.11781	0.14779	0.16159	0.16581	0.17452	0.18000
b_{22}	0.33897	0.34961	0.33398	0.28949	0.30600	0.31368	0.32000
d_1	0.22074	0.23681	0.27335	0.23483	0.26236	0.24807	0.25000
c_1	0.09390	0.00243	−0.02843	−0.03617	−0.04542	−0.05863	−0.05000
δ (%)	22.69146	12.79087	9.13286	6.12151	4.77364	2.56271	

Table 5
The RPEM estimates and errors.

t	100	200	500	1000	2000	3000	True values
a_1	−0.10208	−0.19349	−0.29993	−0.22634	−0.31969	−0.55512	−0.35000
a_2	−0.42407	−0.29942	−0.25924	−0.36924	−0.31144	−0.21561	−0.30000
b_{11}	−0.02274	0.11301	0.03268	0.21966	0.21131	0.13349	0.15000
b_{12}	0.44482	0.39198	0.22399	0.48646	0.44988	0.26059	0.35000
b_{21}	0.50479	0.35918	0.24055	0.12165	0.27222	0.06474	0.18000
b_{22}	0.54137	0.36032	0.25736	0.26767	0.35053	0.30879	0.32000
d_1	−0.15619	−0.31321	0.07434	0.59613	0.28080	0.25085	0.25000
c_1	−0.51747	−0.60803	−0.40152	0.13538	−0.01604	0.03666	−0.05000
δ (%)	108.30440	111.26722	59.27708	60.48342	21.73267	37.50330	

The inputs $u_1(t)$ and $u_2(t)$ are taken as two uncorrelated persistent excitation signal sequences with zero mean and unit variance, and $v(t)$ is taken as an uncorrelated stochastic noise which obeys the normal distribution with zero mean and variance $\sigma^2 = 0.50^2$, the corresponding noise-to-signal ratio of the system is $\delta_{ns} = 67.266\%$. Applying the RGELS, F-RELS, F-ML-RLS and RPEM algorithms to estimate the parameters of this MISO system, the parameter estimates and their errors are shown in Tables 2–5 and the estimation errors $\delta := \|\hat{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ versus t are shown in Fig. 3, the parameter estimates versus t of the F-ML-RLS algorithm are shown in Fig. 4.

From Tables 2–5 and Figs. 3 and 4, we can draw some conclusions as follows.

Please cite this article as: F. Chen, F. Ding, The filtering based maximum likelihood recursive least squares estimation for multiple-input single-output systems, Applied Mathematical Modelling (2015), http://dx.doi.org/10.1016/j.apm.2015.08.015

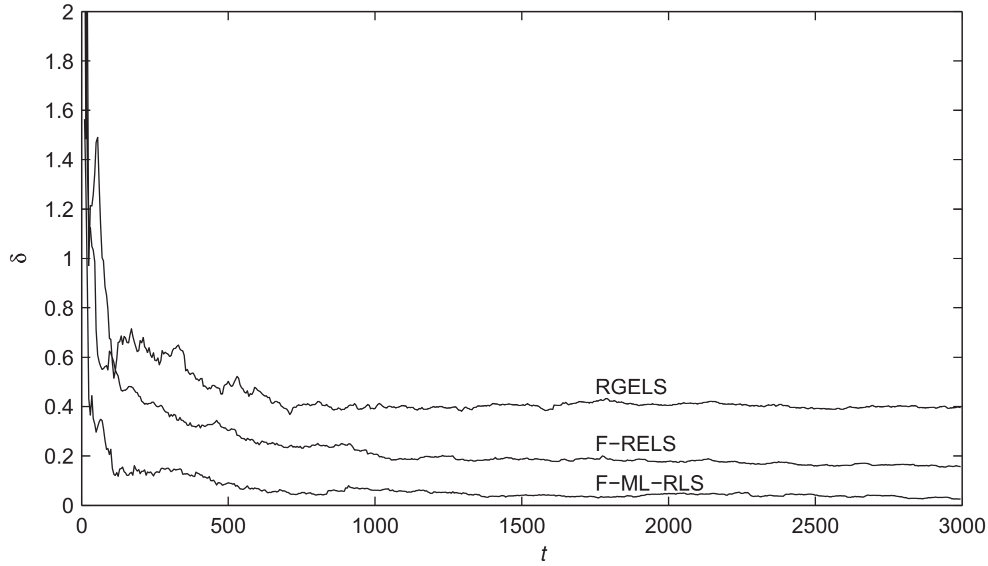

Fig. 3. The parameter estimation errors δ versus t with different algorithms.

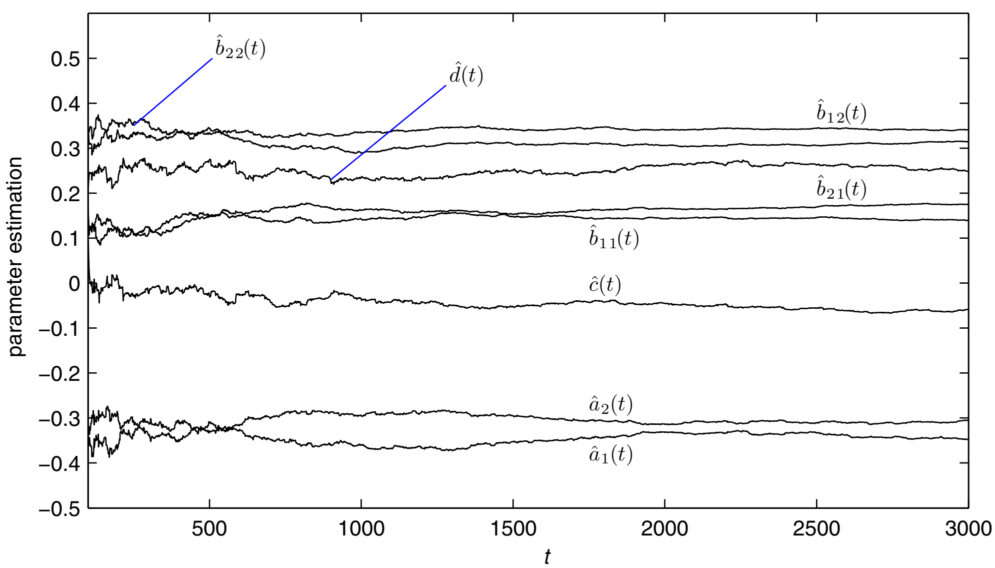

Fig. 4. The F-ML-RLS parameter estimation versus t

- Although the F-ML-RLS algorithm has a relatively lower computational efficiency than the F-RELS algorithm, the parameter estimation accuracy of the F-ML-RLS algorithm is higher than the F-RELS algorithm – see the computational efficiency in Table 1.
- It is clear that the F-ML-RLS algorithm can produce more accurate estimates than the RGELS algorithm and the F-RELS algorithm – see the parameter estimation error curves in Fig. 3.
- As the data length increases, the F-ML-RLS parameter estimates are very close to the their true values – see Fig. 4. This shows that the proposed algorithm has good asymptotic properties.
- The parameter estimation errors of the RGELS, F-RELS, F-ML-RLS and RPEM algorithms become smaller and smaller with the data length t increasing – see the parameter estimates and errors of the last columns in Tables 2–4.
- The parameter estimation accuracy of the F-ML-RLS algorithm is higher than that of the RPEM algorithm – see the estimates and errors in Tables 4 and 5.

6. Conclusions

Based on the maximum likelihood principle and the data filtering technique, we propose three parameter estimation algorithms for multiple-input single-output systems. The proposed algorithms have the following properties.

- The parameter estimation errors given by the three proposed algorithms become generally smaller as the data length increases.
- The proposed F-ML-RLS algorithm can give more accurate parameter estimates and requires lower computational load compared with the RGELS, F-RELS and RPEM algorithms.
- Though the maximum likelihood approach has relatively large calculated quantity, the data filtering technique can reduce the amount of calculation. Hence, the F-ML-RLS algorithm is effective.
- Although the algorithms in the paper are developed for MISO systems, they can be extended to study the identification problems of other multiple-input multiple-output linear or nonlinear systems with colored noise [41–44].

References

[1] M. Hajarian, M. Dehghan, The reflexive and hermitian reflexive solutions of the generalized Sylvester-conjugate matrix equation, Bull. Belg. Math. Soc-Sim. 20 (4) (2013) 639–653.
[2] M. Dehghan, M. Hajarian, The (R,S)-symmetric and (R,S)-skew symmetric solutions of the pair of matrix equations $a_1xb_1 = c_1$ and $a_2xb_2 = c_2$, Bull. Iran. Math. Soc. 37 (3) (2011) 269–279.
[3] B. Hashemi, M. Dehghan, Efficient computation of enclosures for the exact solvents of a quadratic matrix equation, Electron. J. Linear Algebra 20 (2010) 519–536.
[4] F. Ding, Computation of matrix exponentials of special matrices, Appl. Math. Comput. 223 (2013) 311–326.
[5] M. Dehghan, M. Hajarian, Analysis of an iterative algorithm to solve the generalized coupled Sylvester matrix equations, Appl. Math. Modell. 35 (7) (2011) 3285–3300.
[6] B. Hashemi, M. Dehghan, Results concerning interval linear systems with multiple right-hand sides and the interval matrix equation $ax = b$, J. Comput. Appl. Math. 235 (9) (2011) 2969–2978.
[7] M. Dehghani-Madiseh, M. Dehghan, Generalized solution sets of the interval generalized Sylvester matrix equation $\sum_{i=1}^{p} a_i x_i + \sum_{j=1}^{q} y_j b_j = c$ and some approaches for inner and outer estimations, Comput. Math. Appl. 68 (12) (2014) 1758–1774.
[8] G. Mercère, L. Bako, Parameterization and identification of multivariable state-space systems: a canonical approach, Automatica 47 (8) (2011) 1547–1555.
[9] Y. Fu, T.Y. Chai, Nonlinear multivariable adaptive control using multiple models and neural networks, Automatica 43 (6) (2007) 1101–1110.
[10] Y. Gu, F. Ding, J.H. Li, States based iterative parameter estimation for a state space model with multi-state delays using decomposition, Signal Process. 106 (2015) 294–300.
[11] Y. Zhang, Unbiased identification of a class of multi-input single-output systems with correlated disturbances using bias compensation methods, Math. Comput. Modell. 53 (9–10) (2011) 1810–1819.
[12] F.Y. Chen, F. Ding, Recursive least squares identification algorithms for multiple-input nonlinear Box-Jenkins systems using the maximum likelihood principle, J. Comput. Nonlinear Dyn. 11 (2) (2016) 021005–021005-7, doi:10.1115/1.4030387.
[13] Y. Ji, X.M. Liu, New criteria for the robust impulsive synchronization of uncertain chaotic delayed nonlinear systems, Nonlinear Dyn. 79 (1) (2015) 1–9.
[14] J. Vörös, Recursive identification of Hammerstein systems with discontinuous nonlinearities containing dead-zones, IEEE Trans. Autom. Control 48 (12) (2003) 2203–2206.
[15] Y. Shi, H. Fang, Kalman filter based identification for systems with randomly missing measurements in a network environment, Int. J. Control 83 (3) (2010) 538–551.
[16] Y. Shi, B. Yu, Robust mixed H_2/H_infinity control of networked control systems with random time delays in both forward and backward communication links, Automatica 47 (4) (2011) 754–760.
[17] B. Yu, Y. Shi, H. Huang, l-2 and l-infinity filtering for multirate systems using lifted models, Circuits Syst. Signal Process. 27 (5) (2008) 699–711.
[18] Y.W. Mao, F. Ding, Multi-innovation stochastic gradient identification for Hammerstein controlled autoregressive systems based on the filtering technique, Nonlinear Dynamics 79 (3) (2015) 1745–1755.
[19] J. Vörös, Modeling and parameter identification of systems with multi-segment piecewise-linear characteristics, IEEE Trans. Autom. Control 47 (1) (2002) 184–188.
[20] H. Zhang, Y. Shi, A.S. Mehr, Robust H-infinity PID control for multivariable networked control systems with disturbance/noise attenuation, Int. J. Robust Nonlinear Control 22 (2) (2012) 183–204.
[21] Y. Ji, X.M. Liu, Unified synchronization criteria for hybrid switching-impulsive dynamical networks, Circuits Syst. Signal Process. 34 (5) (2015) 1499–1517.
[22] I.D. Landau, R. Lozano, M. M'Saad, A. Karimi, Adaptive Control, Springer, Berlin, 1998.
[23] E.W. Bai, M. Fu, A blind approach to Hammerstein model identification, IEEE Trans. Signal Process. 50 (7) (2002) 1610–1619.
[24] A. Efremov, Recursive estimation of dynamic time-varying demand models, International Conference on Computer Systems and Technologies, Veliko Tarnovo, Bulgaria, June 2006.
[25] F. Ding, X.M. Liu, H.B. Chen, G.Y. Yao, Hierarchical gradient based and hierarchical least squares based iterative parameter identification for CARARMA systems, Signal Process. 97 (2014) 31–39.
[26] F. Ding, Y.J. Wang, J. Ding, Recursive least squares parameter identification for systems with colored noise using the filtering technique and the auxiliary model, Digital Signal Process. 37 (2015) 100–108.
[27] D.Q. Wang, H.B. Liu, Highly efficient identification methods for dual-rate Hammerstein systems, IEEE Trans. Control Syst. Technol. 23 (5) (2015) 1952–1960.
[28] M. Hajarian, M. Dehghan, The generalized centro-symmetric and least squares generalized centro-symmetric solutions of the matrix equation $ayb + cy^T d = e$, Math. Methods Appl. Sci. 34 (13) (2011) 1562–1579.
[29] J.C. Agüero, C.R. Rojas, H. Hjalmarsson, G.C. Goodwin, Accuracy of linear multiple-input multiple-output (MIMO) models obtained by maximum likelihood estimation, Automatica 48 (4) (2012) 632–637.
[30] T. Chen, U.M. Braga-Neto, Maximum-likelihood estimation of the discrete coefficient of determination in stochastic boolean systems, IEEE Trans. Signal Process. 61 (15) (2013) 3880–3894.
[31] B.L. Pence, H.K. Fathy, J.L. Stein, Recursive maximum likelihood parameter estimation for state space systems using polynomial chaos theory, Automatica 47 (11) (2011) 2420–2424.
[32] T. Söderström, M. Hong, J. Schoukens, R. Pintelon, Accuracy analysis of time domain maximum likelihood method and sample maximum likelihood method for errors-in-variables and output error identification, Automatica 46 (4) (2010) 721–727.
[33] L. Vanbeylen, R. Pintelon, J. Schoukens, Blind maximum likelihood identification of Hammerstein systems, Automatica 44 (12) (2008) 3139–3146.
[34] F.Y. Chen, F. Ding, J.H. Li, Maximum likelihood gradient-based iterative estimation algorithm for a class of input nonlinear controlled autoregressive ARMA systems, Nonlinear Dyn. 79 (2) (2015) 927–936.
[35] D.Q. Wang, F. Ding, D.Q. Zhu, Data filtering based least squares algorithms for multivariable CARAR-like systems, Int. J. Control Autom. Syst. 11 (4) (2013) 711–717.

[36] C. Wang, T. Tang, Recursive least squares estimation algorithm applied to a class of linear-in-parameters output error moving average systems, Appl. Math. Lett. 29 (2014) 36–41.
[37] J. Vörös, Identification of Hammerstein systems with time-varying piecewise-linear characteristics, IEEE Trans. Circuits Syst. II Express Briefs 52 (12) (2005) 865–869.
[38] D.T.W. Yau, E.H.K. Fung, H.H.T. Liu, Multivariable identification and controller design of an integrated flight control system, Appl. Math. Modell. 31 (12) (2007) 2733–2743.
[39] T. Söderström, P. Stoica, System Identification, Prentice Hall, New York, 1989.
[40] L. Ljung, System Identification: Theory for the User, second ed., Prentice Hall, Englewood Cliffs, New Jersey, 1999.
[41] F. Ding, K.P. Deng, X.M. Liu, Decomposition based Newton iterative identification method for a Hammerstein nonlinear FIR system with ARMA noise, Circuits Syst. Signal Process. 33 (9) (2014) 2881–2893.
[42] H.B. Chen, Y.S. Xiao, F. Ding, Hierarchical gradient parameter estimation algorithm for Hammerstein nonlinear systems using the key term separation principle, Appl. Math. Comput. 247 (2014) 1202–1210.
[43] D.Q. Zhu, Q. Liu, Z. Hu, Fault-tolerant control algorithm of the manned submarine with multi-thruster based on quantum behaved particle swarm optimization, Int. J. Control 84 (11) (2012) 1817–1829.
[44] D.Q. Zhu, H. Huang, S.X. Yang, Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in 3D underwater workspace, IEEE Trans. Cybern. 43 (2) (2013) 504–514.