Nonlinear Dyn (2014) 76:1195–1202
DOI 10.1007/s11071-013-1202-3

ORIGINAL PAPER

Newton iterative identification method for an input nonlinear finite impulse response system with moving average noise using the key variables separation technique

Kepo Deng · Feng Ding

Received: 13 July 2013 / Accepted: 13 December 2013 / Published online: 3 January 2014

Abstract This paper studies parameter identification problems for input nonlinear finite impulse response systems with moving average noise (i.e., input nonlinear finite impulse response moving average systems). Since the identification model of the system contains the product of the parameters of the nonlinear part and the linear part, we use the key variables separation technique and express the output of the system as the linear combination of all parameters, and then derive a Newton iterative identification method. The simulation results show that the proposed algorithm is effective.

Keywords Iterative identification · Parameter estimation · Newton iteration · Key variables separation · Input nonlinear system

1 Introduction

System identification is to find a model that is close to a real system by using measured data [1–3], and is basic for signal processing, adaptive control, and filtering [4–6]. Nonlinearities exist widely in various aspects of society [7,8], such as engineering practice [9–11], chemical processes [12], and biological systems [13]. The identification of nonlinear systems widely ranges from the structure to the biomedical engineering [14,15] and employs a number of classic and modern approaches [16]. Hammerstein models, Wiener models, and their combinations are the common block-oriented nonlinear models [17–21]. The Hammerstein system consists of a nonlinear static block followed by a linear dynamic subsystem [22]. Recently, Ding and Chen [23] proposed a recursive extended least squares algorithm and a least squares based iterative identification algorithm for Hammerstein ARMAX systems, and a coupled least squares identification methods for multivariate systems [24].

The iterative algorithms are widely used to find the solutions of matrix equations [25,26] and can be used for parameter estimation [27–30]. Recently, Li [31] proposed the maximum likelihood Newton iterative algorithm for Hammerstein CARARMA systems. Ding et al. [32] derived a Newton iterative identification algorithm for Hammerstein nonlinear systems.

The identification model of Hammerstein systems contains the product of the parameters of the nonlinear part and the linear part. Due to this difficulty, Vörös [33,34] proposed the key variables separation technique for Hammerstein systems with discontinuous nonlinearities containing dead-zones [35]. Wang et al. [36] derived the auxiliary model-based recursive

K. Deng · F. Ding (✉)
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

K. Deng
e-mail:dkb1215@163.com

F. Ding
Control Science and Engineering Research Center, Jiangnan University, Wuxi 214122, People's Republic of China

generalized least squares parameter estimation algorithm for Hammerstein OEAR systems. Li and Ding [37] presented a maximum likelihood stochastic gradient algorithm for Hammerstein systems with colored noise based on the key term separation technique.

This paper studies the iterative algorithm for input nonlinear finite impulse response moving average systems and derives a Newton iterative algorithm. By using the key variables separation technique, the parameters of the nonlinear part and the linear part can be directly estimated without using the over-parameterization methods.

The rest of this paper is organized as follows. Section 2 describes the identification model of the Hammerstein finite impulse response moving average systems. Sections 3 and 4 derive the Newton iterative algorithm. Section 5 provides an example to show the effectiveness of the proposed algorithm. Finally, some concluding remarks are offered in Sect. 6.

2 System description

Let us introduce some notation. $\hat{\boldsymbol{\vartheta}}(t)$ denotes the estimate of $\boldsymbol{\vartheta}$ at time t; $\hat{\boldsymbol{\vartheta}}_k$ denotes the estimate of $\boldsymbol{\vartheta}$ at iteration k; $\mathbf{1}_n$ represents an n-dimensional column vector whose elements are 1; the norm of a matrix $\boldsymbol{X}$ is defined by $\|\boldsymbol{X}\|^2 := \mathrm{tr}[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}]$; and the superscript T denotes the matrix transpose.

Consider an input nonlinear finite impulse response moving average (IN-FIR-MA) system in Fig. 1 [37], which consists of a nonlinear static block $f(\cdot)$ followed by a linear finite impulse response moving average (FIR-MA) subsystem, where $u(t)$ is the input sequence of the system, $y(t)$ is the output sequence, $v(t)$ is a white noise with zero mean, and $x(t)$ and $w(t)$ are the inner variables. The output $\bar{u}(t)$ of the nonlinear block is a linear combination of a known basis $\boldsymbol{f}(u(t)) := (f_1(u(t)), f_2(u(t)), \ldots, f_{n_c}(u(t)))$ with coefficients $(c_1, c_2, \ldots, c_{n_c})$ and can be written as

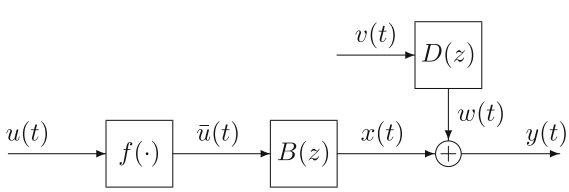

Fig. 1 An input nonlinear finite impulse response moving average system

$$\begin{aligned}\bar{u}(t) &= f(u(t)) = c_1 f_1(u(t)) + c_2 f_2(u(t)) \\ &\quad + \cdots + c_{n_c} f_{n_c}(u(t)) = \sum_{j=1}^{n_c} c_j f_j(u(t)). \end{aligned} \tag{1}$$

The linear part is an FIR-MA model, and $B(z)$ and $D(z)$ are polynomials in the shift operator z^{-1} with

$$B(z) := b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b},$$
$$D(z) := 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_{n_d} z^{-n_d}.$$

Define parameter vectors

$$\boldsymbol{b} := [b_0, b_1, b_2, \ldots, b_{n_b}]^{\mathrm{T}} \in \mathbb{R}^{n_b+1},$$
$$\boldsymbol{c} := [b_1, b_2, \ldots, b_{n_c}]^{\mathrm{T}} \in \mathbb{R}^{n_c},$$
$$\boldsymbol{d} := [d_1, d_2, \ldots, d_{n_d}]^{\mathrm{T}} \in \mathbb{R}^{n_d}.$$

Define the information matrix $\boldsymbol{F}(t)$ and the noise information vector $\boldsymbol{\psi}(t)$ as

$$\begin{aligned}\boldsymbol{F}(t) &:= [\boldsymbol{f}(u(t)), \boldsymbol{f}(u(t-1)), \\ &\qquad \ldots, \boldsymbol{f}(u(t-n_b))]^{\mathrm{T}} \in \mathbb{R}^{n_b \times n_c},\\ \boldsymbol{\psi}(t) &:= [v(t-1), v(t-2), \ldots, v(t-n_d)]^{\mathrm{T}} \in \mathbb{R}^{n_d}.\end{aligned}$$

Then the output $y(t)$ in Fig. 1 can be expressed as

$$\begin{aligned}y(t) &= x(t) + w(t)\\ &= B(z)\bar{u}(t) + D(z)v(t)\\ &= (b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b})\bar{u}(t)\\ &\quad + (1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_{n_d} z^{-n_d})v(t)\\ &= b_0\bar{u}(t) + b_1\bar{u}(t-1) + b_2\bar{u}(t-2)\\ &\quad + \cdots + b_{n_b}\bar{u}(t-n_b) + d_1 v(t-1)\\ &\quad + d_2 v(t-2) + \cdots + d_{n_d} v(t-n_d) + v(t) \qquad (2)\\ &= b_0\boldsymbol{f}(u(t))\boldsymbol{c} + b_1\boldsymbol{f}(u(t-1))\boldsymbol{c} + b_2\boldsymbol{f}(u(t-2))\boldsymbol{c}\\ &\quad + \cdots + b_{n_b}\boldsymbol{f}(u(t-n_b))\boldsymbol{c} + \boldsymbol{\psi}^{\mathrm{T}}(t)\boldsymbol{d} + v(t)\\ &= \boldsymbol{b}^{\mathrm{T}}\boldsymbol{F}(t)\boldsymbol{c} + \boldsymbol{\psi}^{\mathrm{T}}(t)\boldsymbol{d} + v(t). \qquad (3)\end{aligned}$$

Equation (3) contains the product of the parameters $\boldsymbol{b}$ and $\boldsymbol{c}$, so it is difficult to identify the parameters of the system. In order to avoid the parameter product of the linear and nonlinear blocks, we use the key variables separation technique and let the coefficient $b_0 = 1$ [31]. Then Eq. (2) can be rewritten as

$$\begin{aligned}y(t) &= \bar{u}(t) + b_1\bar{u}(t-1) + b_2\bar{u}(t-2)\\ &\quad + \cdots + b_{n_b}\bar{u}(t-n_b) + \boldsymbol{\psi}^{\mathrm{T}}(t)\boldsymbol{d} + v(t).\end{aligned} \tag{4}$$

Here the first term $\bar{u}(t)$ on the right-hand side of (4) is chosen as a separated key variable, and the rests are taken as the non-separated key variables. Referring to the key variables separation principle [33,34], substituting $\bar{u}(t)$ in (1) into the separated key variable

$\bar{u}(t)$ in (4) and keeping the non-separated key variables unchanged give

$$\begin{aligned} y(t) &= c_1 f_1(u(t)) + c_2 f_2(u(t)) + \cdots + c_{n_c} f_{n_c}(u(t)) \\ &\quad + b_1\bar{u}(t-1) + b_2\bar{u}(t-2) \\ &\quad + \cdots + b_{n_b}\bar{u}(t-n_b) + \boldsymbol{\psi}^{\mathrm{T}}(t)\boldsymbol{d} + v(t). \end{aligned} \tag{5}$$

Using the key variables separation technique, we express the output $y(t)$ of the system as the linear regressive form of all parameters—see Eq. (5).

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

$$\begin{aligned} \boldsymbol{\vartheta} &:= [b_1, b_2, \ldots, b_{n_b}, c_1, c_2, \ldots, c_{n_c}, d_1, d_2, \\ &\qquad \ldots, d_{n_d}]^{\mathrm{T}} \in \mathbb{R}^{n_b+n_c+n_d}, \\ \boldsymbol{\varphi}(t) &:= [\bar{u}(t-1), \bar{u}(t-2), \ldots, \\ &\qquad \bar{u}(t-n_b), f_1(u(t)), f_2(u(t)), \ldots, f_{n_c}(u(t)), \\ &\qquad v(t-1), v(t-2), \ldots, v(t-n_d)]^{\mathrm{T}} \\ &\quad \in \mathbb{R}^{n_b+n_c+n_d}. \end{aligned}$$

Thus, Eq. (5) can be rewritten as

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\vartheta} + v(t). \tag{6}$$

3 The Newton iterative algorithm

Consider a set of data from $i = t - L + 1$ to $i = t$ (L represents the data length). Define the stacked output vector $\boldsymbol{Y}(t)$ and the stacked information matrices $\boldsymbol{\Phi}(t)$ as

$$\begin{aligned} \boldsymbol{Y}(t) &:= \begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-L+1) \end{bmatrix} \in \mathbb{R}^{L}, \\ \boldsymbol{\Phi}(t) &:= \begin{bmatrix} \boldsymbol{\varphi}^{\mathrm{T}}(t) \\ \boldsymbol{\varphi}^{\mathrm{T}}(t-1) \\ \vdots \\ \boldsymbol{\varphi}^{\mathrm{T}}(t-L+1) \end{bmatrix} \\ &\quad \in \mathbb{R}^{L\times(n_b+n_c+n_d)}. \end{aligned} \tag{7}$$

Define the criterion function,

$$J(\boldsymbol{\vartheta}) := \|\boldsymbol{Y}(t) - \boldsymbol{\Phi}(t)\boldsymbol{\vartheta}\|^2.$$

The gradient of $J(\boldsymbol{\vartheta})$ with respect to $\boldsymbol{\vartheta}$ is

$$\begin{aligned} \mathrm{grad}[J(\boldsymbol{\vartheta})] &= -2\boldsymbol{\Phi}^{\mathrm{T}}(t)[\boldsymbol{Y}(t) - \boldsymbol{\Phi}(t)\boldsymbol{\vartheta}] \\ &\qquad \in \mathbb{R}^{(n_b+n_c+n_d)\times(n_b+n_c+n_d)}. \end{aligned}$$

Compute the Hessian matrix of the cost function $J(\boldsymbol{\vartheta})$ with respect to $\boldsymbol{\vartheta}$:

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{\vartheta}) : &= \frac{\partial^2 J(\boldsymbol{\vartheta})}{\partial\boldsymbol{\vartheta}\partial\boldsymbol{\vartheta}^{\mathrm{T}}} = \frac{\partial \mathrm{grad}[J(\boldsymbol{\vartheta})]}{\partial\boldsymbol{\vartheta}^{\mathrm{T}}} \\ &= 2\boldsymbol{\Phi}^{\mathrm{T}}(t)\boldsymbol{\Phi}(t) \in \mathbb{R}^{(n_b+n_c+n_d)\times(n_b+n_c+n_d)}. \end{aligned}$$

Using the Newton method and minimizing $J(\boldsymbol{\vartheta})$ give

$$\begin{aligned} \hat{\boldsymbol{\vartheta}}_k(t) &= \hat{\boldsymbol{\vartheta}}_{k-1}(t) - [\boldsymbol{H}(\hat{\boldsymbol{\vartheta}}_{k-1}(t))]^{-1}\mathrm{grad}[J(\hat{\boldsymbol{\vartheta}}_{k-1}(t))] \\ &= \hat{\boldsymbol{\vartheta}}_{k-1}(t) + [\boldsymbol{\Phi}^{\mathrm{T}}(t)\boldsymbol{\Phi}(t)]^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(t) \\ &\quad \times[\boldsymbol{Y}(t) - \boldsymbol{\Phi}(t)\hat{\boldsymbol{\vartheta}}_{k-1}(t)]. \end{aligned} \tag{8}$$

Because the information matrix $\boldsymbol{\varphi}(t)$ contains the unknown inner variables $\bar{u}(t-i)$ (the output of non-linear block) and $v(t-i)$, the above algorithm cannot be applied to estimate $\boldsymbol{\vartheta}$. The solution is to use the auxiliary model identification idea [38]: the unknown variables are replaced with the outputs of the auxiliary model, the unknown variables $\bar{u}(t-i)$ are replaced with their corresponding estimate $\hat{\bar{u}}_{k-1}(t-i)$ at iteration $k-1$, and $v(t-i)$ are replaced with their estimates $\hat{v}(t-i)$. Define

$$\begin{aligned} \hat{\boldsymbol{\varphi}}_k(t) &:= [\hat{\bar{u}}_{k-1}(t-1), \hat{\bar{u}}_{k-1}(t-2), \\ &\qquad \ldots, \hat{\bar{u}}_{k-1}(t-n_b), \boldsymbol{f}(u(t)), \hat{v}_{k-1}(t-1), \\ &\qquad \hat{v}_{k-1}(t-2), \ldots, \hat{v}_{k-1}(t-n_d)]^{\mathrm{T}} \\ &\qquad \in \mathbb{R}^{n_b+n_c+n_d}. \end{aligned}$$

Let the iterative estimate $\hat{\boldsymbol{\vartheta}}_k(t)$ of $\boldsymbol{\vartheta}$ at iteration k be

$$\begin{aligned} \hat{\boldsymbol{\vartheta}}_k(t) &:= [\hat{b}_{1,k}(t), \hat{b}_{2,k}(t), \ldots, \hat{b}_{n_b,k}(t), \hat{\boldsymbol{c}}_k^{\mathrm{T}}(t), \hat{\boldsymbol{d}}_k^{\mathrm{T}}(t)]^{\mathrm{T}} \\ &\qquad \in \mathbb{R}^{n_b+n_c+n_d}, \\ \hat{\boldsymbol{c}}_k(t) &:= [\hat{c}_{1,k}(t), \hat{c}_{2,k}(t), \ldots, \hat{c}_{n_c,k}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_c}, \\ \hat{\boldsymbol{d}}_k(t) &:= [\hat{d}_{1,k}(t), \hat{d}_{2,k}(t), \ldots, \hat{d}_{n_d,k}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_d}. \end{aligned}$$

Substituting c_j in (1) with $\hat{c}_{j,k}(t)$, the iterative estimate $\hat{\bar{u}}_k(t-i)$ of $\bar{u}(t-i)$ at iteration k can be compute through

$$\begin{aligned} \hat{\bar{u}}_k(t-i) &= \hat{c}_{1,k}(t)f_1(u(t-i)) + \hat{c}_{2,k}(t)f_2(u(t-i)) \\ &\qquad + \cdots + \hat{c}_{m,k}(t)f_m(u(t-i)) \\ &= \sum_{j=1}^{m}\hat{c}_{j,k}(t)f_j(u(t-i)) \\ &= \boldsymbol{f}(u(t-i))\hat{\boldsymbol{c}}_k(t). \end{aligned}$$

The estimates $\hat{v}_k(t-i)$ of $v(t-i)$ can be compute through

$$\hat{v}_k(t-i) = y(t-i) - \hat{\boldsymbol{\varphi}}_k(t-i)\hat{\boldsymbol{\vartheta}}_k(t).$$

Replace $\boldsymbol{\varphi}(t)$ in $\boldsymbol{\Phi}(t)$ with $\hat{\boldsymbol{\varphi}}_k(t)$, and define

$$\hat{\boldsymbol{\Phi}}_k(t) := [\hat{\boldsymbol{\varphi}}_k(t), \hat{\boldsymbol{\varphi}}_k(t-1), \ldots, \hat{\boldsymbol{\varphi}}_k(t-L+1)]^{\mathrm{T}} \in \mathbb{R}^{L\times(n_b+n_c+n_d)}.$$

Substituting $\boldsymbol{\Phi}(t)$ in (8) with $\hat{\boldsymbol{\Phi}}_k(t)$, we can obtain the Newton iterative algorithm for input nonlinear finite impulse response systems:

$$\begin{aligned}\hat{\boldsymbol{\vartheta}}_k(t) &= \hat{\boldsymbol{\vartheta}}_{k-1}(t) + [\hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\Phi}}_k(t)]^{-1}\hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(t) \\ &\quad\times[\boldsymbol{Y}(t) - \hat{\boldsymbol{\Phi}}_k(t)\hat{\boldsymbol{\vartheta}}_{k-1}(t)] \qquad (9)\\ &= [\hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\Phi}}_k(t)]^{-1}\hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(t)\boldsymbol{Y}(t),\ k = 1, 2, 3, \ldots \qquad (10)\end{aligned}$$

$$\boldsymbol{Y}(t) = [y(t), y(t-1), \ldots, y(t-L+1)]^{\mathrm{T}}, \qquad (11)$$

$$\hat{\boldsymbol{\Phi}}_k(t) = [\hat{\boldsymbol{\varphi}}_k(t), \hat{\boldsymbol{\varphi}}_k(t-1), \ldots, \hat{\boldsymbol{\varphi}}_k(t-L+1)]^{\mathrm{T}}, \qquad (12)$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}_k(j) &= [\hat{\bar{u}}_{k-1}(j-1), \hat{\bar{u}}_{k-1}(j-2), \\ &\quad \ldots, \hat{\bar{u}}_{k-1}(j-n_b), \boldsymbol{f}(u(j)), \hat{v}_{k-1}(j-1), \\ &\quad \hat{v}_{k-1}(j-2), \ldots, \hat{v}_{k-1}(j-n_d)]^{\mathrm{T}}, \\ &\quad j = t-L+1, t-L+2, \ldots, t, \qquad (13)\end{aligned}$$

$$\hat{\bar{u}}_k(j-i) = \boldsymbol{f}(u(j-i))\hat{\boldsymbol{c}}_k(t),\ i = 1, 2, \ldots, n_b, \qquad (14)$$

$$\hat{v}_k(j-i) = y(j-i) - \hat{\boldsymbol{\varphi}}_k(j-i)\hat{\boldsymbol{\vartheta}}_k(t),\ i = 1, 2, \ldots, n_d, \qquad (15)$$

$$\boldsymbol{f}(u(j)) = [f_1(u(j)), f_2(u(j)), \ldots, f_m(u(j))], \qquad (16)$$

$$\begin{aligned}\hat{\boldsymbol{\vartheta}}_k(t) &= [\hat{b}_{1,k}(t), \hat{b}_{2,k}(t), \ldots, \hat{b}_{n_b,k}(t), \hat{\boldsymbol{c}}_k^{\mathrm{T}}(t), \\ &\quad \hat{d}_{1,k}(t), \hat{d}_{2,k}(t), \ldots, \hat{d}_{n_d,k}(t)]^{\mathrm{T}}, \qquad (17)\end{aligned}$$

$$\hat{\boldsymbol{c}}_k(t) = [\hat{c}_{1,k}(t), \hat{c}_{2,k}(t), \ldots, \hat{c}_{n_c,k}(t)]^{\mathrm{T}}. \qquad (18)$$

The procedure for computing the parameter estimation vector $\hat{\boldsymbol{\vartheta}}_k(t)$ in the Newton iterative algorithm in (9)–(18) is as follows:

1. Set the data length L, let $t = L$. Collect the input–output data $\{u(i), y(i): i = 0, 1, 2, \ldots, L-1\}$, and pre-set a small $\varepsilon > 0$.
2. Collect the input–output data $u(t)$ and $y(t)$ and form $\boldsymbol{Y}(t)$ by using (11).
3. Let $k = 1$, and set the initial values $\hat{\boldsymbol{\vartheta}}_0(t) = \mathbf{1}_{n_b+n_c+n_d}/p_0$, $p_0 = 10^6$.
4. Form $\hat{\boldsymbol{\varphi}}_k(j)$ by using (13), and construct $\hat{\boldsymbol{\Phi}}_k(t)$ by using (12).
5. Update the parameter estimate $\hat{\boldsymbol{\vartheta}}_k(t)$ by using (9).
6. Compute $\hat{\bar{u}}_k(j-i)$ by using (14), and $\hat{v}_k(j-i)$ by using (15).
7. If $\|\hat{\boldsymbol{\vartheta}}_k(t) - \hat{\boldsymbol{\vartheta}}_{k-1}(t)\| > \varepsilon$, increase k by 1 and go to step 4; otherwise, obtain k and $\hat{\boldsymbol{\theta}}_k(t)$, let $\hat{\boldsymbol{\vartheta}}_0(t+1) = \hat{\boldsymbol{\theta}}_k(t)$, and increase t by 1 and go to step 2.

The flowchart of computing the parameter estimation vector $\hat{\boldsymbol{\vartheta}}_k(t)$ is shown in Fig. 2.

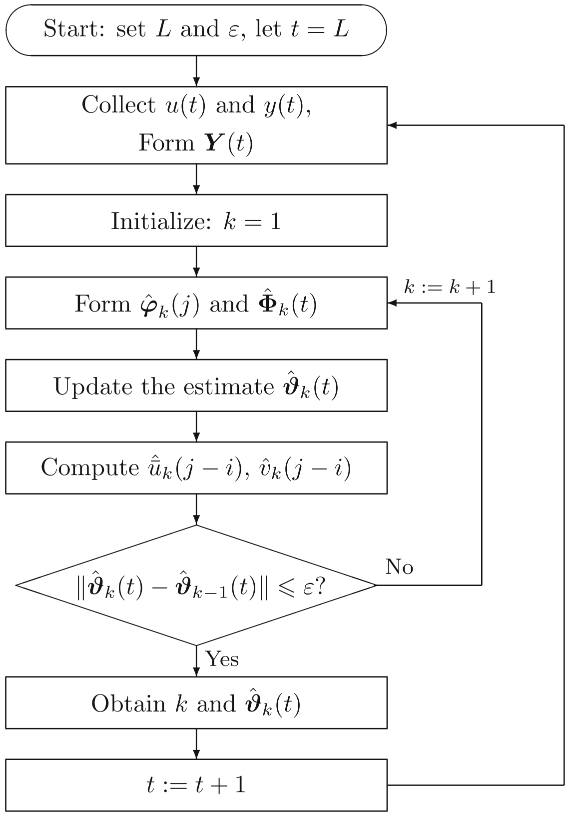

Fig. 2 The flowchart of computing the Newton iteration parameter estimate $\hat{\boldsymbol{\vartheta}}_k(t)$

4 The Newton iterative algorithm with finite measurement data

Let L represents the data length, set $p = L$ and $t = L$ in (7). Then we have

$$\boldsymbol{Y} := \begin{bmatrix} y(L) \\ y(L-1) \\ \vdots \\ y(1) \end{bmatrix} \in \mathbb{R}^L,$$

$$\boldsymbol{\Phi} := \begin{bmatrix} \boldsymbol{\varphi}^{\mathrm{T}}(L) \\ \boldsymbol{\varphi}^{\mathrm{T}}(L-1) \\ \vdots \\ \boldsymbol{\varphi}^{\mathrm{T}}(1) \end{bmatrix} \in \mathbb{R}^{L\times(n_b+n_c+n_d)}.$$

$\boldsymbol{Y}$ and $\boldsymbol{\Phi}$ contain all of the input–output data $\{u(t), y(t) : t = 1, 2, \ldots, L\}$. The criterion function is defined as

$J_1(\boldsymbol{\vartheta}) := \|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\vartheta}\|^2$.

The unknown variables $\bar{u}(t-i)$ in the information matrix $\boldsymbol{\Phi}$ are replaced with their corresponding estimate $\hat{\bar{u}}_{k-1}(t-i)$ at iteration $k-1$, and $v(t-i)$ are replaced with their estimates $\hat{v}(t-i)$. Similarly, we can obtain the Newton iterative algorithm for the input nonlinear finite impulse response system with finite measurement data:

$$\hat{\boldsymbol{\vartheta}}_k = \hat{\boldsymbol{\vartheta}}_{k-1} + (\hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}\hat{\boldsymbol{\Phi}}_k)^{-1}\hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(\boldsymbol{Y} - \hat{\boldsymbol{\Phi}}_k\hat{\boldsymbol{\vartheta}}_{k-1}) \tag{19}$$

$$= (\hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}\hat{\boldsymbol{\Phi}}_k)^{-1}\hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}\boldsymbol{Y},\ k = 1, 2, 3, \ldots \tag{20}$$

$$\boldsymbol{Y} = [y(L), y(L-1), \ldots, y(1)]^{\mathrm{T}}, \tag{21}$$

$$\hat{\boldsymbol{\Phi}}_k = [\hat{\boldsymbol{\varphi}}_k(L), \hat{\boldsymbol{\varphi}}_k(L-1), \ldots, \hat{\boldsymbol{\varphi}}_k(1)]^{\mathrm{T}}, \tag{22}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}_k(t) = {} & [\hat{\bar{u}}_{k-1}(t-1), \hat{\bar{u}}_{k-1}(t-2), \\ & \ldots, \hat{\bar{u}}_{k-1}(t-n_b), \boldsymbol{f}(u(t)), \hat{v}_{k-1}(t-1), \\ & \hat{v}_{k-1}(t-2), \\ & \ldots, \hat{v}_{k-1}(t-n_d)]^{\mathrm{T}}, t = 1, 2, \ldots, L,\end{aligned} \tag{23}$$

$$\hat{\bar{u}}_k(t) = \boldsymbol{f}(u(t))\hat{\boldsymbol{c}}_k, \tag{24}$$

$$\hat{v}_k(t) = y(t) - \hat{\boldsymbol{\varphi}}_k(t)\hat{\boldsymbol{\vartheta}}_k, \tag{25}$$

$$\boldsymbol{f}(u(t)) = [f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t))], \tag{26}$$

$$\hat{\boldsymbol{\vartheta}}_k = [\hat{b}_{1,k}, \hat{b}_{2,k}, \ldots, \hat{b}_{n_b,k}, \hat{\boldsymbol{c}}_k^{\mathrm{T}}, \hat{d}_{1,k}, \hat{d}_{2,k}, \ldots, \hat{d}_{n_d,k}]^{\mathrm{T}}, \tag{27}$$

$$\hat{\boldsymbol{c}}_k = [\hat{c}_{1,k}, \hat{c}_{2,k}, \ldots, \hat{c}_{n_c,k}]^{\mathrm{T}}. \tag{28}$$

The procedure for computing the parameter estimation vector $\hat{\boldsymbol{\vartheta}}_k$ in (19)–(28) is as follows:

1. Set the data length L, and pre-set a small $\varepsilon > 0$. Collect the input–output data $\{u(t),\ y(t)$: $t = 1, 2, \cdots, L\}$, form $\boldsymbol{Y}$ by using (21).
2. Let $k = 1$, and set the initial values $\hat{\boldsymbol{\vartheta}}_0 = \mathbf{1}_{n_b+n_c+n_d}/p_0$, $p_0 = 10^6$.
3. Form $\hat{\boldsymbol{\varphi}}_k(t)$ by using (23), and construct $\hat{\boldsymbol{\Phi}}_k$ by using (22).
4. Update the parameter estimate $\hat{\boldsymbol{\vartheta}}_k$ by using (19).
5. Compute $\hat{\bar{u}}_k(t)$ by using (24), and $\hat{v}_k(t)$ by using (25).
6. If $\|\hat{\boldsymbol{\vartheta}}_k - \hat{\boldsymbol{\vartheta}}_{k-1}\| > \varepsilon$, increase k by 1 and go to step 3; otherwise, obtain k and $\hat{\boldsymbol{\vartheta}}_k$.

The flowchart of computing the parameter estimation vector $\hat{\boldsymbol{\vartheta}}_k$ is shown in Fig. 3.

By using a batch of input–output data to update the parameter estimates, the Newton iterative algorithm has faster convergence rates compared with the gradient-based iterative algorithm in [27], although the Newton iterative algorithms require computing the Hessian matrix and the matrix inversion. If the input–output data are sufficiently rich, then this matrix inversion exists.

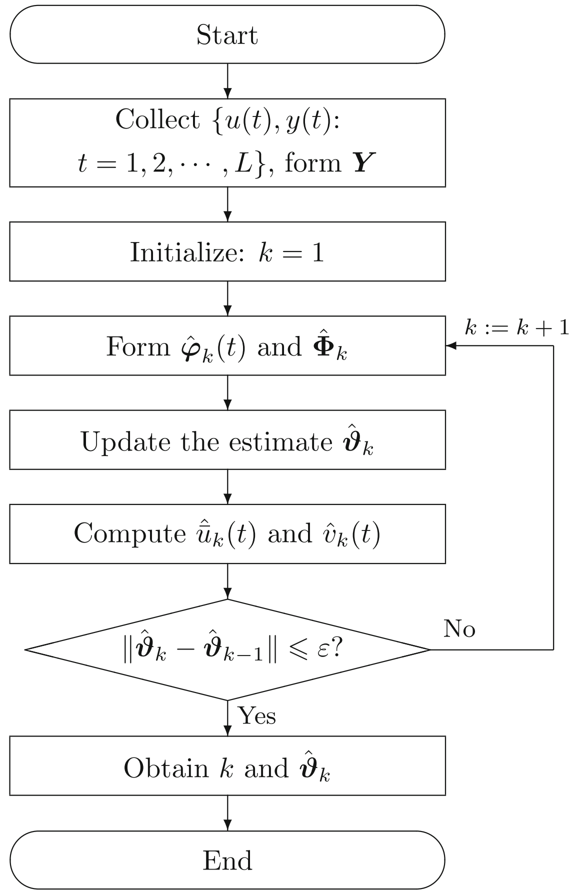

Fig. 3 The flowchart of computing the parameter estimate $\hat{\boldsymbol{\vartheta}}_k$

5 Example

Consider the following nonlinear system:

$$y(t) = B(z)\bar{u}(t) + D(z)v(t),$$

$$B(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} = 1 + 0.85z^{-1} + 0.65z^{-2},$$

$$D(z) = 1 + d_1 z^{-1} = 1 + 0.40z^{-1},$$

$$\bar{u}(t) = c_1 u(t) + c_2 u^2(t) = 0.80u(t) + 0.50u^2(t),$$

$$\boldsymbol{\vartheta} = [b_1, b_2, c_1, c_2, d_1]^{\mathrm{T}} = [0.85, 0.65, 0.80, 0.50, 0.40]^{\mathrm{T}}.$$

In simulation, the input $u(t)$ is taken as an uncorrelated stochastic signal sequence with zero mean and unit variance, and $v(t)$ as a white noise sequence with zero mean and variances $\sigma^2 = 0.10^2$ and $\sigma^2 = 0.50^2$. Taking the date length $L = 2000$, and applying the proposed Newton iterative algorithm in (19)–(28) to esti-

Table 1 The parameter estimates and errors ($\sigma^2 = 0.10^2$, $L = 2000$)

k	b_1	b_2	c_1	c_2	d_1	$\delta(\%)$
1	−0.02125	0.01437	0.79608	0.91720	0.00541	82.46951
2	0.64427	0.29479	0.79482	0.39709	0.21823	31.08906
3	0.90262	0.65401	0.80610	0.53468	0.17450	15.81112
4	0.83911	0.66071	0.80179	0.49392	0.42339	1.93365
5	0.85349	0.64678	0.80257	0.50192	0.39470	0.52653
6	0.84983	0.64882	0.80253	0.49888	0.39515	0.38527
7	0.85102	0.64852	0.80258	0.49984	0.39685	0.30064
8	0.85064	0.64862	0.80257	0.49956	0.39642	0.31591
9	0.85075	0.64858	0.80257	0.49964	0.39655	0.31105
10	0.85072	0.64859	0.80257	0.49962	0.39651	0.31267
True values	0.85000	0.65000	0.80000	0.50000	0.40000	

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

k	b_1	b_2	c_1	c_2	d_1	$\delta(\%)$
1	−0.00995	0.03471	0.80176	0.91430	0.00586	81.13927
2	0.66446	0.26302	0.80394	0.39694	0.26629	31.12856
3	0.89242	0.61925	0.81551	0.51882	0.28986	8.39560
4	0.84501	0.64952	0.81216	0.49151	0.38721	1.36491
5	0.85630	0.64227	0.81291	0.50027	0.39721	1.11703
6	0.85271	0.64332	0.81282	0.49748	0.39632	1.03705
7	0.85376	0.64292	0.81286	0.49832	0.39658	1.05505
8	0.85344	0.64302	0.81285	0.49806	0.39649	1.04938
9	0.85354	0.64298	0.81285	0.49814	0.39652	1.05116
10	0.85351	0.64299	0.81285	0.49812	0.39651	1.05062
True values	0.85000	0.65000	0.80000	0.50000	0.40000	

mate the parameters of this example system, the parameter estimates and their errors $\delta := \|\hat{\boldsymbol{\vartheta}}_k - \boldsymbol{\vartheta}\|/\|\boldsymbol{\vartheta}\|$ are shown in Tables 1, 2 and Fig. 4.

From Tables 1, 2, and Fig. 4, we can draw the following conclusions. (1) The estimation errors are small for iteration $k \geq 5$, the parameter estimates oscillate, because there exist disturbance noises. (2) The parameter estimates are very close to their true values for large k. (3) A lower noise level results in a smaller parameter estimation error.

Furthermore, using the Monte Carlo simulations with 20 sets of noise realizations, the parameter estimates and the estimation variances of the Newton iterative algorithms are shown in Tables 3 and 4 with $\sigma^2 = 0.10^2$, $\sigma^2 = 0.50^2$, and $L = 2000$. From Tables 3 and 4, we can see that the average values of the parameter estimates are very close to the true parameters, and the variances are small for iteration $k > 5$. This validates the performance of the proposed Newton iterative algorithm.

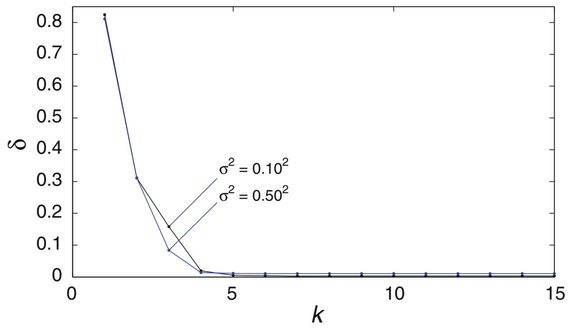

Fig. 4 The parameter estimation errors versus k with different variances

Table 3 The parameter estimates and variances based on the 20 Monte Carlo runs ($\sigma^2 = 0.10^2$)

k	b_1	b_2	c_1	c_2	d_1
1	-0.02308 ± 0.00931	0.01006 ± 0.00494	0.79478 ± 0.00332	0.91764 ± 0.00427	0.00471 ± 0.00290
2	0.64284 ± 0.00294	0.29512 ± 0.00940	0.79296 ± 0.00328	0.39701 ± 0.00403	0.21861 ± 0.00599
3	0.90237 ± 0.00379	0.65512 ± 0.00437	0.80421 ± 0.00335	0.53425 ± 0.00342	0.17747 ± 0.01772
4	0.83908 ± 0.00327	0.66322 ± 0.00405	0.79925 ± 0.00330	0.49403 ± 0.00299	0.43059 ± 0.01130
5	0.85372 ± 0.00351	0.64891 ± 0.00494	0.79985 ± 0.00320	0.50177 ± 0.00197	0.39152 ± 0.00498
6	0.85013 ± 0.00342	0.65117 ± 0.00515	0.79985 ± 0.00315	0.49873 ± 0.00209	0.39924 ± 0.00570
7	0.85131 ± 0.00345	0.65086 ± 0.00509	0.79989 ± 0.00315	0.49970 ± 0.00210	0.39956 ± 0.00547
8	0.85093 ± 0.00345	0.65097 ± 0.00510	0.79988 ± 0.00316	0.49942 ± 0.00209	0.39948 ± 0.00556
9	0.85104 ± 0.00345	0.65093 ± 0.00509	0.79988 ± 0.00315	0.49950 ± 0.00210	0.39950 ± 0.00553
10	0.85101 ± 0.00345	0.65094 ± 0.00510	0.79988 ± 0.00315	0.49948 ± 0.00209	0.39950 ± 0.00553
True values	0.85000	0.65000	0.80000	0.50000	0.40000

Table 4 The parameter estimates and variances based on the 20 Monte Carlo runs ($\sigma^2 = 0.50^2$)

k	b_1	b_2	c_1	c_2	d_1
1	-0.01909 ± 0.04652	0.01317 ± 0.02466	0.79525 ± 0.01660	0.91652 ± 0.02133	0.00236 ± 0.01451
2	0.65658 ± 0.01288	0.26613 ± 0.03262	0.79394 ± 0.01700	0.39589 ± 0.01932	0.26550 ± 0.01927
3	0.89258 ± 0.01833	0.62579 ± 0.02452	0.80425 ± 0.01621	0.51747 ± 0.00986	0.29847 ± 0.03094
4	0.84499 ± 0.01685	0.66026 ± 0.02442	0.79937 ± 0.01591	0.49119 ± 0.01196	0.39383 ± 0.00791
5	0.85750 ± 0.01777	0.65299 ± 0.02535	0.79948 ± 0.01581	0.49960 ± 0.01037	0.39967 ± 0.00491
6	0.85415 ± 0.01751	0.65476 ± 0.02564	0.79936 ± 0.01575	0.49677 ± 0.01058	0.39944 ± 0.00564
7	0.85529 ± 0.01757	0.65451 ± 0.02563	0.79940 ± 0.01576	0.49764 ± 0.01053	0.39951 ± 0.00554
8	0.85495 ± 0.01754	0.65463 ± 0.02562	0.79939 ± 0.01576	0.49739 ± 0.01057	0.39949 ± 0.00555
9	0.85505 ± 0.01755	0.65460 ± 0.02562	0.79939 ± 0.01575	0.49746 ± 0.01056	0.39950 ± 0.00554
10	0.85502 ± 0.01755	0.65461 ± 0.02561	0.79939 ± 0.01575	0.49744 ± 0.01056	0.39949 ± 0.00555
True values	0.85000	0.65000	0.80000	0.50000	0.40000

6 Conclusions

This paper proposes a Newton iterative algorithm for the input nonlinear finite impulse response moving average system. The output of the system can be expressed as the linear regressive form of all parameters by using the key variables separation technique. The simulation results indicate that the proposed algorithms have fast convergence rates and accurate estimates compared with the gradient- based iterative algorithm. The proposed algorithm can be combined with other identification methods [39,40] to study identification problems of other linear or nonlinear systems with colored noises.

Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 61273194), the Fundamental Research Funds for the Central Universities (No. JUSRP51322B), the PAPD of Jiangsu Higher Education Institutions and the 111 Project (B12018).

References

1. Ljung, L.: System Identification: Theory for the User, 2nd edn. Prentice Hall, Englewood Cliffs (1999)
2. Goodwin, G.C., Sin, K.S.: Adaptive Filtering Prediction and Control. Prentice Hall, Englewood Cliffs (1984)
3. Ding, F.: System Identification—New Theory and Methods. Science, Beijing (2013)

4. Zhu, D.Q., Kong, M.: Adaptive fault-tolerant control of nonlinear system: an improved CMAC based fault learning approach. Int. J. Control **80**(10), 1576–1594 (2007)
5. Zhu, D.Q., Gu, W.: Sensor fusion for integrated circuit fault diagnosis using a belief function model. Int. J. Distrib. Sens. Netw. **6**(4), 247–261 (2008)
6. Zhu, D.Q., Liu, Q., Yang, Y.S.: An active fault-tolerant control method of unmanned underwater vehicles with continuous and uncertain faults. Int. J. Adv. Robotic Syst. **5**(4), 411–418 (2008)
7. Ikeda, T., Harata, Y., Ibrahim, R.A.: Nonlinear liquid sloshing in square tanks subjected to horizontal random excitation. Nonlinear Dyn. **72**(1–2), 439–453 (2013)
8. Rashid, M.T., Frasca, M., et al.: Nonlinear model identification for Artemia population motion. Nonlinear Dyn. **69**(4), 2237–2243 (2012)
9. 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)
10. Shi, Y., Yu, B.: 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), 754–760 (2011)
11. Zhang, Q.J., Luo, J., Wan, L.: Parameter identification and synchronization of uncertain general complex networks via adaptive-impulsive control. Nonlinear Dyn. **71**(1–2), 353–359 (2013)
12. Liu, L.C., Tian, B., Xue, Y.S., et al.: Analytic solution for a nonlinear chemistry system of ordinary differential equations. Nonlinear Dyn. **68**(1–2), 17–21 (2012)
13. Chakraborty, K., Haldar, S., Kar, T.K.: Global stability and bifurcation analysis of a delay induced prey–predator system with stage structure. Nonlinear Dyn. **73**(3), 1307–1325 (2013)
14. Kerschen, G., Worden, K., Vakakis, A.F., Golinval, J.C.: Past, present and future of nonlinear system identification in structural dynamics. Mech. Syst. Signal Process. **20**(3), 505–592 (2006)
15. Giannakis, G., Serpedin, E.: A bibliography on nonlinear system identification. Signal Process. **81**(3), 533–580 (2001)
16. Nelles, O.: Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Springer, Berlin (2001)
17. Wills, A., Schön, T.B., Ljung, L., Ninness, B.: Identification of Hammerstein–Wiener models. Automatica **49**(1), 70–81 (2013)
18. Wang, Z.Y., Shen, Y.X., Ji, Z.C., et al.: Filtering based recursive least squares algorithm for Hammerstein FIR-MA systems. Nonlinear Dyn. **73**(1–2), 1045–1054 (2013)
19. Wang, D.Q., Ding, F.: Least squares based and gradient based iterative identification for Wiener nonlinear systems. Signal Process. **91**(5), 1182–1189 (2011)
20. Wang, D.Q., Ding, F.: Hierarchical least squares estimation algorithm for Hammerstein–Wiener systems. IEEE Signal Process. Lett. **19**(12), 825–828 (2012)
21. Hu, P.P., Ding, F.: Multistage least squares based iterative estimation for feedback nonlinear systems with moving average noises using the hierarchical identification principle. Nonlinear Dyn. **73**(1–2), 583–592 (2013)
22. Wang, D.Q., Ding, F., Chu, Y.Y.: Data filtering based recursive least squares algorithm for Hammerstein systems using the key-term separation principle. Inf. Sci. **222**, 203–212 (2013)
23. Ding, F., Chen, T.: Identification of Hammerstein nonlinear ARMAX systems. Automatica **41**(9), 1479–1489 (2005)
24. Ding, F.: Coupled-least-squares identification for multivariable systems. IET Control Theory Appl. **7**(1), 68–79 (2013)
25. Dehghan, M., Hajarian, M.: Iterative algorithms for the generalized centro-symmetric and central anti-symmetric solutions of general coupled matrix equations. Eng. Comput. **29**(5), 528–560 (2012)
26. Dehghan, M., Hajarian, M.: Fourth-order variants of Newton's method without second derivatives for solving nonlinear equations. Eng. Comput. **29**(4), 356–365 (2012)
27. Ding, F., Liu, X.G., Chu, J.: Gradient-based and least-squares-based iterative algorithms for Hammerstein systems using the hierarchical identification principle. IET Control Theory Appl. **7**(2), 176–184 (2013)
28. 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)
29. Ding, F.: Two-stage least squares based iterative estimation algorithm for CARARMA system modeling. Appl. Math. Model. **37**(7), 4798–4808 (2013)
30. Ding, F., Liu, X.M., Chen, H.B., Yao, G.Y.: Hierarchical gradient based and hierarchical least squares based iterative parameter identification for CARARMA systems. Signal Process. **97**, 31–39 (2014)
31. Li, J.H.: Parameter estimation for Hammerstein CARARMA systems based on the Newton iteration. Appl. Math. Lett. **26**(1), 91–96 (2013)
32. Ding, F., Liu, X.P., Liu, G.: Identification methods for Hammerstein nonlinear systems. Digit. Signal Process. **21**(2), 215–238 (2011)
33. Vörös, J.: Modeling and parameter identification of systems with multi-segment piecewise-linear characteristics. IEEE Trans. Autom. Control **47**(1), 184–188 (2002)
34. Vörös, J.: Recursive identification of Hammerstein systems with discontinuous nonlinearities containing dead-zones. IEEE Trans. Autom. Control **48**(12), 2203–2206 (2003)
35. Shen, Q.Y., Ding, F.: Iterative estimation methods for Hammerstein controlled autoregressive moving average systems based on the key-term separation principle. Nonlinear Dyn. (2014). doi:10.1007/s11071-013-1097-z
36. Wang, D.Q., Chu, Y.Y., et al.: Auxiliary model based recursive generalized least squares parameter estimation for Hammerstein OEAR systems. Math. Comput. Model. **52**(1–2), 309–317 (2010)
37. Li, J.H., Ding, F.: Maximum likelihood stochastic gradient estimation for Hammerstein systems with colored noise based on the key term separation technique. Comput. Math. Appl. **62**(11), 4170–4177 (2011)
38. Ding, F.: Decomposition based fast least squares algorithm for output error systems. Signal Process. **93**(5), 1235–1242 (2013)
39. Ding, F.: Combined state and least squares parameter estimation algorithms for dynamic systems. Appl. Math. Model. **38**(1), 403–412 (2014)
40. Liu, Y.J., Sheng, J., Ding, R.F.: Convergence of stochastic gradient estimation algorithm for multivariable ARX-like systems. Comput. Math. Appl. **59**(8), 2615–2627 (2010)