Parameter Identification of the Input Nonlinear Systems with the Colored Noise

Jingfan Liu*, Xiangqun Li, Darong Gao

School of Electrical Engineering, Northwest University for Nationalities, China

Abstract

Nonlinear systems widely exist in practical applications, like communication systems, chemical processes, biomedical systems and so on. Therefore, nonlinear systems identificati Jingfan Liu on is quite significant both in theory and application. This thesis presents the identification algorithms for a class of nonlinear systems based on the Youth Project of Central University. Considering the identification of the input nonlinear systems with the colored noise, An extended Newton recursive algorithm are derived for comparison. In the simulation, the results show that the Newton recursive algorithm can get better accurate parameter estimates, The simulation results show the effectiveness of the proposed algorithms.

Keywords

Newton recursion, Newton method, input nonlinear systems, Hammerstein models

1 Introduction

System identification plays an important role in system modeling, for example,signal processingand control engineering [1,3]. Parameter estimation is basic for system modeling and analysis [3-5]. Its purpose is to estimate system parameters according to a certain criterion function by using the observed input and output data. There are many effective methods in identifying systems such as the gradient search based algorithms [6–9], least squares approximation [10–13]. System identification is advancing at a fast speed, and new ideas and methods are emerging in many fields. This paper mainly studies the identification of the Hammerstein nonlinear systems with color noise. The system is widely used in practice, for example, the description of PH value, the process with nonlinear characteristics such as curtain function, dead zone, switching and so on. Therefore, it is of great significance to study this kind of time series model.

Many scholars have done a lot of work on the parameter identification problem of Hammerstein system. Narendra and Gallman proposed a selection identification algorithm for nonlinear systems, referred to as NG algorithm [14], Stoica pointed out that there was a convergence problem in this algorithm [15]. Later, Rangan, Wolodkin and Poolla proved that when the linear part of the Hammerstein system was a finite impulse response model (FIR) and the system input was white noise, NG algorithm converges [16]. Chang and Luus proposed to identify Hammerstein system with colored noise by generation selection algorithm, but failed to prove the convergence of the algorithm [17]. Ba,Cerone and Regruto deduced that $D(z)/A(z)=1$ (**Figure 1**) Hammerstein model output error is bounded, then parameters are also bounded [18]. In recent years, Bai proved the convergence of the generation selection algorithm in succession, but the generation selection algorithm wasn't suitable for online identification [19]. Ding and Chen proposed a recursive method of Hammerstein system that could be used for online identification by using the principle of least squares, and proved its convergence by using the martingale convergence method [20,21]. Vanbeylen, Pintelon and Schoukens researched the maximum likelihood identification method for Hammerstein systems with Gaussian noise, invertible nonlinearity and zero output error [22]. Giri, Rochdi, Chaoui and Brouri identified the

AIoTC2022@International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology
EMAIL: * Corresponding author: e-mail address: liujingfan_2006@126.com (Jingfan Liu)

linear and nonlinear subsystems of the hysteresis Hammerstein system respectively using least squares [23]. Xiang Wei and Zonghai Chen proposed a new identification method for Volterra sequence (Laguerre function), a specific dynamic model of the nonlinear system. Many new methods are also used to identify nonlinear systems such as neural networks and genetic algorithms [24,25].

2 System description and identification model

Hammerstein model nonlinear system is the input nonlinear system, it consists of a static nonlinear segment and a dynamic linear segment(**Figure 1**) [26], where $y(t)$ is output, $u(t)$ is input, $\overline{u}(t)$ is the output of the nonlinear part and the noise $v(t)$ is assumed to be i.i.d. random sequences with zero mean, $A(z)$, $B(z)$ and $D(z)$ are polynomials in the unit backward shift operator $z^{-1}[z^{-1}y(t)=y(t-1)]$, with

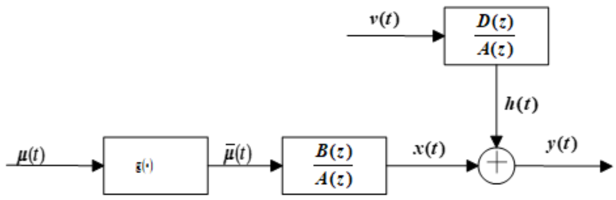

Figure 1 Hammerstein system

$$A(z)=1+\alpha_1 z^{-1}+\alpha_2 z^{-2}+\cdots+\alpha_{n_a} z^{-n_a},$$
$$B(z)=\beta_1 z^{-1}+\beta_2 z^{-2}+\beta_3 z^{-3}+\cdots+\beta_{n_b} z^{-n_b},$$
$$D(z)=1+d_1 z^{-1}+d_2 z^{-2}+\cdots+d_{n_d} z^{-n_d}.$$

The intermediate variables $\overline{\mu}(t)$, $x(t)$ and $h(t)$ are immesurable, and $g(\cdot)$ is static nonlinear function of state. The nonlinear part is an unknown polynomial that can be expressed as [27]:

$$\begin{aligned}\overline{\mu}(t)&=\boldsymbol{g}(\mu(t))\\&=c_1 g_1(\mu(t))+c_2 g_2(\mu(t))+\cdots+c_{n_c} g_{n_c}(\mu(t))\\&=\boldsymbol{g}(\mu(t))\boldsymbol{c}\end{aligned} \tag{1}$$

We can write the Hammerstein-CARMA model (**Figure 1**) into the following formula:

$$A(z)y(t)=B(z)\overline{\mu}(t)+D(z)v(t) \tag{2}$$

The unknown parameters needed to be estimated are: the linear subsystem parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{d}$ and the nonlinear part parameters $\boldsymbol{c}$ .Let the superscript T represent the matrix transpose.

$$\boldsymbol{\alpha}:=[\alpha_1,\alpha_2,\cdots,\alpha_{n_a}]^{\mathrm{T}}\in\mathbb{R}^{n_a},\qquad \boldsymbol{\beta}:=[\beta_1,\beta_2,\cdots,\beta_{n_b}]^{\mathrm{T}}\in\mathbb{R}^{n_b},$$
$$\boldsymbol{d}:=[d_1,d_2,\cdots,d_{n_d}]^{\mathrm{T}}\in\mathbb{R}^{n_d},$$
$$\boldsymbol{\gamma}:=\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{d}\end{bmatrix}\in\mathbb{R}^{n_a+n_d},$$
$$\boldsymbol{\theta}:=\begin{bmatrix}\boldsymbol{\gamma}\\ \boldsymbol{\beta}\\ \boldsymbol{c}\end{bmatrix}\in\mathbb{R}^{n},n:=n_a+n_b+n_c+n_d.$$

Let $\hat{\boldsymbol{\theta}}(t):=\begin{bmatrix}\hat{\boldsymbol{\gamma}}_t\\ \hat{\boldsymbol{\beta}}_t\\ \hat{\boldsymbol{c}}_t\end{bmatrix}$ denote the estimate of $\boldsymbol{\theta}:=\begin{bmatrix}\boldsymbol{\gamma}\\ \boldsymbol{\beta}\\ \boldsymbol{c}\end{bmatrix}$ at time t .Define the information vector $\boldsymbol{\varphi}(t)$ as:

$$\boldsymbol{\varphi}(t) := [-y(t-1), -y(t-2), \cdots, -y(t-n_a),\\ v(t-1),\ v(t-2), \cdots, v(t-n_d)]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_d} \tag{3}$$

The output sequence can be written as:

$$y(t) = \boldsymbol{\zeta}^{\mathrm{T}}(t)\boldsymbol{J} + v(t) \tag{4}$$

Thereinto $\boldsymbol{J}$ is the parameterized vector, $\boldsymbol{\zeta}(t)$ is the information vector, and they are defined as:

$$\boldsymbol{J} := \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \otimes \boldsymbol{c} \end{bmatrix} \in \mathbb{R}^{n_a+n_b n_c+n_d},$$

$$\boldsymbol{\zeta}(t) := [\boldsymbol{\varphi}^{\mathrm{T}}(t), \boldsymbol{g}(\mu(t-1)), \boldsymbol{g}(\mu(t-2)), \cdots\\ \boldsymbol{g}(\mu(t-n_b))]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_b n_c+n_d}.$$

$\boldsymbol{\beta} \otimes \boldsymbol{c}$ is the Kronecker product of $\boldsymbol{\beta}$ and $\boldsymbol{c}$. In most of the existing papers, the combined parameter $\boldsymbol{\beta} \otimes \boldsymbol{c}$ is identified, and the combined parameter needs to be decomposed after the identification result is obtained [28], which increases the computational burden. The goal of this paper is to identify the parameters through the extended Newton recursive algorithm, obtain the parameter vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{c}$ and $\boldsymbol{d}$.

3 The extended Newton recursive algorithm

3.1 The algorithm description

In this section, Newton method will be used to derive the augmented Newton recursive identification algorithm based on Hammerstein-CARMA model, its basic idea is to introduce stacking output vector and stacking information matrix. Define the input information:

$$\boldsymbol{G}(t) := \begin{bmatrix} \boldsymbol{g}(\mu(t-1)) \\ \boldsymbol{g}(\mu(t-2) \\ \vdots \\ \boldsymbol{g}(\mu(t-n_b)) \end{bmatrix} \in \mathbb{R}^{n_b \times n_c} \tag{5}$$

From Eq.(1) and Eq. (2), we have:

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\gamma} + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t)\boldsymbol{c} + v(t) \tag{6}$$

We define a quadratic criterion function as follows:

$$J_1(\boldsymbol{\theta}) = J_1(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{c}) = [y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\gamma} - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t)\boldsymbol{c}]^2 \tag{7}$$

Since the Hessian matrix $\boldsymbol{H}[J_1(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{c})]$ of the criterion function J_1 is singular, it is useful to introduce the stacked data, so Newton algorithm is used to solve the identification optimization problem.

$$\boldsymbol{H}[J_1(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{c})] = 2\begin{bmatrix} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t) & \boldsymbol{\varphi}(t)\boldsymbol{c}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}(t) & \boldsymbol{\varphi}(t)\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t) \\ \boldsymbol{G}(t)\boldsymbol{c}\boldsymbol{\varphi}^{\mathrm{T}}(t) & \boldsymbol{G}(t)\boldsymbol{c}\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{G}^{\mathrm{T}}(t) & \boldsymbol{h}_{23}(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{c}, t) \\ \boldsymbol{G}^{\mathrm{T}}(t)\boldsymbol{\beta}\boldsymbol{\varphi}^{\mathrm{T}}(t) & \boldsymbol{h}_{32}(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{c}, t) & \boldsymbol{G}^{\mathrm{T}}(t)\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t) \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Where:

$$\begin{aligned}\boldsymbol{h}_{23}(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{c}, t) :&= -\frac{\partial}{\partial \boldsymbol{c}}\{\boldsymbol{G}(t)\boldsymbol{c}[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\gamma} + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t)\boldsymbol{c}]\} \\ &= -\boldsymbol{G}(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\gamma} + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t)\boldsymbol{c}] \\ &\quad + \boldsymbol{G}(t)\boldsymbol{c}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t) \in \mathbb{R}^{n_b \times n_c}\end{aligned}$$

$$\begin{aligned}\boldsymbol{h}_{32}(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{c}, t) :&= -\frac{\partial}{\partial \boldsymbol{b}}\{\boldsymbol{G}^{\mathrm{T}}(t)\boldsymbol{\beta}[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\gamma} + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t)\boldsymbol{c}]\} \\ &= -\boldsymbol{G}^{\mathrm{T}}(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\gamma} + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t)\boldsymbol{c}] + \boldsymbol{G}^{\mathrm{T}}(t)\boldsymbol{\beta}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}(t) \\ &= \boldsymbol{h}_{23}^{\mathrm{T}}(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{c}, t) \in \mathbb{R}^{n_c \times n_b}\end{aligned}$$

Consider the newest p data and define stacked output vector $\boldsymbol{Y}(p,t)$ and stacked matrices $\boldsymbol{\Phi}_0(p,t)$, $\boldsymbol{\Phi}(\boldsymbol{c},t)$ and $\boldsymbol{\Psi}(\boldsymbol{\beta},t)$.

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

$$\boldsymbol{\Phi}_0(p,t):=\begin{bmatrix} \boldsymbol{\varphi}^{\mathrm{T}}(t) \\ \boldsymbol{\varphi}^{\mathrm{T}}(t\text{-}1) \\ \vdots \\ \boldsymbol{\varphi}^{\mathrm{T}}(t-p+1) \end{bmatrix} \in \mathbb{R}^{p\times(n_a+n_d)}$$

$$\boldsymbol{\Phi}(\boldsymbol{c},t):=\begin{bmatrix} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}(t) \\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}(t\text{-}1) \\ \vdots \\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}(t-p+1) \end{bmatrix} \in \mathbb{R}^{p\times n_b}$$

$$\boldsymbol{\Psi}(\boldsymbol{\beta},t):=\begin{bmatrix} \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t) \\ \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t\text{-}1) \\ \vdots \\ \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t-p+1) \end{bmatrix} \in \mathbb{R}^{p\times n_c}$$

Then define a new criterion function:

$$\begin{aligned} J_2(\boldsymbol{\theta}) &= J_2(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c}) \\ &:= \|\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\boldsymbol{\gamma}-\boldsymbol{\Psi}(\boldsymbol{\beta},t)\boldsymbol{c}\|^2 \\ &= \|\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\boldsymbol{\gamma}-\boldsymbol{\Phi}(\boldsymbol{\beta},t)\boldsymbol{\gamma}\|^2 \end{aligned} \tag{8}$$

Eq.(8) is equivalent to the following criterion function constructed from the data in a dynamical window with length p.

$$J_3(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c}) = \sum_{i=t-p+1}^{t} [y(i)-\boldsymbol{\varphi}^{\mathrm{T}}(i)\boldsymbol{\gamma}-\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(i)\boldsymbol{c}]^2 \tag{9}$$

That is $J_2(\boldsymbol{\theta}) = J_3(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c})$. If we take $t=N$ and $p=N$ (N is the data length), then Eq.(8) and Eq.(9) are the least squares criterion functions [29].

Computing the gradient of $J_2(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c})$ gives:

$$\begin{aligned} grad_{\theta}[J_2(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c})] &= -2\begin{bmatrix} \boldsymbol{\Phi}_0^{\mathrm{T}}(p,t) \\ \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{c},t) \\ \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\beta},t) \end{bmatrix}[\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\boldsymbol{\gamma}-\boldsymbol{\Psi}(\boldsymbol{\beta},t)\boldsymbol{c}] \\ &= -2\begin{bmatrix} \boldsymbol{\Phi}_0^{\mathrm{T}}(p,t) \\ \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{c},t) \\ \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\beta},t) \end{bmatrix}[\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\boldsymbol{\gamma}-\boldsymbol{\Phi}(\boldsymbol{c},t)\boldsymbol{\beta}] \end{aligned}$$

Define the extended generalized information matrix $\boldsymbol{\Xi}(t)$ and expanding innovation into innovation vector $\boldsymbol{E}(p,t)$ as:

$$\boldsymbol{\Xi}(t):=\begin{bmatrix} \boldsymbol{\Phi}_0^{\mathrm{T}}(p,t) \\ \boldsymbol{\Phi}^{\mathrm{T}}(\hat{\boldsymbol{c}}_{t-1},t) \\ \boldsymbol{\Psi}^{\mathrm{T}}(\hat{\boldsymbol{\beta}}_{t-1},t) \end{bmatrix} \in \mathbb{R}^{n\times p}$$

$$\begin{aligned}\boldsymbol{E}(p,t)&:=\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\hat{\boldsymbol{\gamma}}_{t-1}-\boldsymbol{\Psi}(\hat{\boldsymbol{\beta}}_{t-1},t)\hat{\boldsymbol{c}}_{t-1}\\&=\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\hat{\boldsymbol{\gamma}}_{t-1}-\boldsymbol{\Phi}(\hat{\boldsymbol{c}}_{t-1},t)\hat{\boldsymbol{\beta}}_{t-1}\in\mathbb{R}^p\end{aligned}\tag{10}$$

Thus, we have:

$$\begin{aligned}grad_\theta[J_2(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c})]&=-2\begin{bmatrix}\boldsymbol{\Phi}_0^{\mathrm{T}}(p,t)\\\boldsymbol{\Phi}^{\mathrm{T}}(\hat{\boldsymbol{c}}_{t-1},t)\\\boldsymbol{\Psi}^{\mathrm{T}}(\hat{\boldsymbol{\beta}}_{t-1},t)\end{bmatrix}\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\hat{\boldsymbol{\gamma}}_{t-1}-\boldsymbol{\Psi}(\hat{\boldsymbol{\beta}}_{t-1},t)\hat{\boldsymbol{c}}_{t-1}\\&=-2\begin{bmatrix}\boldsymbol{\Phi}_0^{\mathrm{T}}(p,t)\\\boldsymbol{\Phi}^{\mathrm{T}}(\hat{\boldsymbol{c}}_{t-1},t)\\\boldsymbol{\Psi}^{\mathrm{T}}(\hat{\boldsymbol{\beta}}_{t-1},t)\end{bmatrix}\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\hat{\boldsymbol{\gamma}}_{t-1}-\boldsymbol{\Phi}(\hat{\boldsymbol{c}}_{t-1},t)\hat{\boldsymbol{\beta}}_{t-1}\\&=-2\boldsymbol{\Xi}(t)\boldsymbol{E}(p,t)\end{aligned}\tag{11}$$

Computing the Hessian matrix of the criterion function $J_2(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c})$.

$$\begin{aligned}\boldsymbol{H}[J_2(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c})]&=\frac{\partial gard_\theta\left[J_2(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c})\right]}{\partial\theta^{\mathrm{T}}}\\&=2\begin{bmatrix}\boldsymbol{\Phi}_0^{\mathrm{T}}(p,t)\boldsymbol{\Phi}_0(p,t)&\boldsymbol{\Phi}_0^{\mathrm{T}}(p,t)\boldsymbol{\Phi}(\boldsymbol{c},t)&\boldsymbol{\Phi}_0^{\mathrm{T}}(p,t)\boldsymbol{\Psi}(\boldsymbol{\beta},t)\\\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{c},t)\boldsymbol{\Phi}_0(p,t)&\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{c},t)\boldsymbol{\Phi}(\boldsymbol{c},t)&\boldsymbol{H}_{23}(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c},t)\\\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\beta},t)\boldsymbol{\Phi}_0(p,t)&\boldsymbol{H}_{23}^{\mathrm{T}}(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c},t)&\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\beta},t)\boldsymbol{\Psi}(\boldsymbol{\beta},t)\end{bmatrix}\in\mathbb{R}^{n\times n}\end{aligned}$$

Where:

$$\begin{aligned}\boldsymbol{H}_{23}(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{c},t)&:=-\frac{\partial}{\partial\boldsymbol{c}}\left\{\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{c},t)[\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\boldsymbol{\gamma}-\boldsymbol{\Psi}(\boldsymbol{\beta},t)\boldsymbol{c}]\right\}\\&=-\frac{\partial}{\partial\boldsymbol{c}}\{[\boldsymbol{G}(t)\boldsymbol{c},\boldsymbol{G}(t\text{-}1)\boldsymbol{c},\cdots,\boldsymbol{G}(t\text{-}p+1)\boldsymbol{c}][\boldsymbol{Y}(p,t)-\boldsymbol{\Phi}_0(p,t)\boldsymbol{\gamma}-\boldsymbol{\Psi}(\boldsymbol{\beta},t)\boldsymbol{c}]\}\\&=-\frac{\partial}{\partial\boldsymbol{c}}\left\{\sum_{i=0}^{p-1}\boldsymbol{G}(t-i)\boldsymbol{c}\left[y(t-i)-\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\boldsymbol{\gamma}-\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t-i)\boldsymbol{c}\right]\right\}\\&=-\sum_{i=0}^{p-1}\left\{\boldsymbol{G}(t-i)\left[y(t-i)-\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\boldsymbol{\gamma}-\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t-i)\boldsymbol{c}\right]-\boldsymbol{G}(t-i)\boldsymbol{c}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t-i)\right\}\\&=\sum_{i=0}^{p-1}\left\{\boldsymbol{G}(t-i)\left[-y(t-i)+\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\boldsymbol{\gamma}+\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t-i)\boldsymbol{c}\right]+\boldsymbol{G}(t-i)\boldsymbol{c}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t-i)\right\}\\&=\sum_{i=0}^{p-1}\left\{\boldsymbol{G}(t-i)\left[-y(t-i)+\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\boldsymbol{\gamma}+\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}(t-i)\boldsymbol{c}\right]\right\}+\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{c},t)\boldsymbol{\Psi}(\boldsymbol{\beta},t)\in\mathbb{R}^{n_b\times n_c}\end{aligned}$$

Using the Newton method to minimize $J_2(\boldsymbol{\theta})$, we can obtain the following recursive relation of computing $\hat{\boldsymbol{\theta}}(t)$:

$$\begin{aligned}\hat{\boldsymbol{\theta}}(t)&=\hat{\boldsymbol{\theta}}(t-1)-\left\{\boldsymbol{H}[J_2(\hat{\boldsymbol{\eta}}_{t-1},\hat{\boldsymbol{b}}_{t-1},\hat{\boldsymbol{c}}_{t-1})]\right\}^{-1}\mathrm{grad}_\theta[J_2(\hat{\boldsymbol{\eta}}_{t-1},\hat{\boldsymbol{b}}_{t-1},\hat{\boldsymbol{c}}_{t-1})]\\&=\hat{\boldsymbol{\theta}}(t-1)+2\left\{\boldsymbol{H}[J_2(\hat{\boldsymbol{\eta}}_{t-1},\hat{\boldsymbol{b}}_{t-1},\hat{\boldsymbol{c}}_{t-1})]\right\}^{-1}\boldsymbol{\Xi}(t)\boldsymbol{E}(p,t)\end{aligned}\tag{12}$$

On the right side of the equation(12) containing the unknown Hessian matrix $\boldsymbol{H}[J_2(\hat{\boldsymbol{\gamma}}_{t-1},\hat{\boldsymbol{\beta}}_{t-1},\hat{\boldsymbol{c}}_{t-1})]$, extended generalized information matrix $\boldsymbol{\Xi}(t)$ and innovation vector $\boldsymbol{E}(p,t)$, and $\boldsymbol{\varphi}(t)$ contains the unpredictable noise $v(t-i),i=1,2,\cdots,n_d$ .In order to solve these difficulties, according to the principle of recursive identification, replacing $\boldsymbol{H}[J_2(\hat{\boldsymbol{\gamma}}_{t-1},\hat{\boldsymbol{\beta}}_{t-1},\hat{\boldsymbol{c}}_{t-1})]$, $\boldsymbol{\Xi}(t)$ and $\boldsymbol{E}(p,t)$ in the above Eq.(12) with $\boldsymbol{H}[J_2(\hat{\boldsymbol{\gamma}}_{t-1},\hat{\boldsymbol{\beta}}_{t-1},\hat{\boldsymbol{c}}_{t-1})]$, $\hat{\boldsymbol{\Xi}}(t)$ and $\hat{\boldsymbol{E}}(p,t)$, let $\hat{v}(t-i)$ denote the estimate of $v(t-i)$ to define the estimate of $\boldsymbol{\varphi}(t)$ as follows:

$$\begin{aligned}\hat{\boldsymbol{\varphi}}(t):=[&-y(t-1),-y(t-2),\cdots,-y(t-n_a),\\&\hat{v}(t-1),\ \hat{v}(t-2),\cdots,\hat{v}(t-n_d)]^{\mathrm{T}}\end{aligned}$$

From Eq.(6), $\hat{v}(t)$ can be written as:

$$\hat{v}(t)=y(t)-\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\gamma}}-\hat{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{G}(t)\hat{\boldsymbol{c}}$$

We can summarize the Newton extended recursive algorithm (the H-ENR algorithm) for the Hammerstein-CARMA models as follows:

$$\hat{\boldsymbol{\theta}}(t)=\hat{\boldsymbol{\theta}}(t-1)+\hat{\Pi}^{-1}(t)\boldsymbol{\Xi}(t)\boldsymbol{E}(p,t)\tag{13}$$

$$\hat{v}(t)=y(t)-\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\gamma}}-\hat{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{G}(t)\hat{\boldsymbol{c}}\tag{14}$$

$$\hat{\boldsymbol{\Pi}}(t)=\frac{1}{2}\left\{\hat{\boldsymbol{H}}[J_2(\hat{\boldsymbol{\gamma}}_{t-1},\hat{\boldsymbol{\beta}}_{t-1},\hat{\boldsymbol{c}}_{t-1})]\right\}$$
$$=\begin{bmatrix} \hat{\boldsymbol{\Phi}}_0^{\mathrm{T}}(p,t)\hat{\boldsymbol{\Phi}}_0(p,t) & \hat{\boldsymbol{\Phi}}_0^{\mathrm{T}}(p,t)\boldsymbol{\Phi}(\boldsymbol{c}_{t-1},t) & \hat{\boldsymbol{\Phi}}_0^{\mathrm{T}}(p,t)\boldsymbol{\Psi}(\boldsymbol{\beta}_{t-1},t) \\ \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{c}_{t-1},t)\hat{\boldsymbol{\Phi}}_0(p,t) & \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{c}_{t-1},t)\boldsymbol{\Phi}(\boldsymbol{c}_{t-1},t) & \hat{\boldsymbol{\Pi}}_{23}(t) \\ \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\beta}_{t-1},t)\hat{\boldsymbol{\Phi}}_0(p,t) & \hat{\boldsymbol{\Pi}}_{23}^{\mathrm{T}}(t) & \boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\beta}_{t-1},t)\boldsymbol{\Psi}(\boldsymbol{\beta}_{t-1},t) \end{bmatrix} \tag{15}$$

$$\hat{\boldsymbol{\Pi}}_{23}(t)=\sum_{i=0}^{p-1}\left\{\boldsymbol{G}(t-i)\left[-y(t-i)+\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t-i)\hat{\boldsymbol{\gamma}}_{t-1}+\hat{\boldsymbol{\beta}}_{t-1}^{\mathrm{T}}\boldsymbol{G}(t-i)\hat{\boldsymbol{c}}_{t-1}\right]\right\}$$
$$+\boldsymbol{\Phi}^{\mathrm{T}}(\hat{\boldsymbol{c}}_{t-1},t)\boldsymbol{\Psi}(\hat{\boldsymbol{\beta}}_{t-1},t) \tag{16}$$

$$\hat{\boldsymbol{\Xi}}(t)=\left[\hat{\boldsymbol{\Phi}}_0^{\mathrm{T}}(p,t) \quad \boldsymbol{\Phi}^{\mathrm{T}}(\hat{\boldsymbol{c}}_{t-1},t) \quad \boldsymbol{\Psi}^{\mathrm{T}}(\hat{\boldsymbol{\beta}}_{t-1},t)\right]^{\mathrm{T}} \tag{17}$$

$$\hat{\boldsymbol{E}}(p,t)=\boldsymbol{Y}(p,t)-\hat{\boldsymbol{\Phi}}_0(p,t)\hat{\boldsymbol{\gamma}}_{t-1}-\boldsymbol{\Phi}(\hat{\boldsymbol{c}}_{t-1},t)\hat{\boldsymbol{\beta}}_{t-1} \tag{18}$$

$$\boldsymbol{Y}(p,t)=\left[y(t),\ y(t-1),\cdots,\ y(t-p+1)\right]^{\mathrm{T}} \tag{19}$$

$$\hat{\boldsymbol{\Phi}}_0(p,t)=\left[\hat{\boldsymbol{\varphi}}(t),\ \hat{\boldsymbol{\varphi}}(t\text{-}1),\cdots,\ \hat{\boldsymbol{\varphi}}(t-p+1)\right]^{\mathrm{T}} \tag{20}$$

$$\boldsymbol{\Psi}(\hat{\boldsymbol{\beta}}_{t-1},t)=\left[\boldsymbol{G}^{\mathrm{T}}(t)\hat{\boldsymbol{\beta}}_{t-1},\boldsymbol{G}^{\mathrm{T}}(t-1)\hat{\boldsymbol{\beta}}_{t-1},\cdots\boldsymbol{G}^{\mathrm{T}}(t-p+1)\hat{\boldsymbol{\beta}}_{t-1}\right]^{\mathrm{T}} \tag{21}$$

$$\boldsymbol{\Phi}(\hat{\boldsymbol{c}}_{t-1},t)=\left[\boldsymbol{G}(t)\hat{\boldsymbol{c}}_{t-1},\boldsymbol{G}(t-1)\hat{\boldsymbol{c}}_{t-1},\cdots,\boldsymbol{G}(t-p+1)\hat{\boldsymbol{c}}_{t-1}\right]^{\mathrm{T}} \tag{22}$$

$$\begin{aligned}\hat{\boldsymbol{\varphi}}(t)=[&-y(t-1),-y(t-2),\cdots,-y(t-n_a),\\ &\hat{v}(t-1),\ \hat{v}(t-2),\cdots,\hat{v}(t-n_d)]^{\mathrm{T}}\end{aligned} \tag{23}$$

$$\hat{\boldsymbol{c}}_t=\operatorname{sgn}[\hat{\boldsymbol{\theta}}_{n_a+n_b+n_d+1}(t)]\frac{[\hat{\boldsymbol{\theta}}(t)](n_a+n_b+n_d+1:n)}{\|[\hat{\boldsymbol{\theta}}(t)](n_a+n_b+n_d+1:n)\|} \tag{24}$$

$$[\hat{\boldsymbol{\theta}}(t)](n_a+n_b+n_d+1:n)=\hat{\boldsymbol{c}}_t \tag{25}$$

The inverse matrix $\hat{\Omega}^{-1}(t)$ in Eq. (13), for all t ,the stacked data length p in the nonsingular matrix $\hat{\Omega}(t)$ should be large enough to invert the nonsingular matrix. The process of computing $\hat{\boldsymbol{\theta}}(t)$ by the H-ENR algorithm is summarised as follows:

(1) Choose the stacked data length p and initialize: let $t=1$, $\hat{\boldsymbol{\theta}}(0)$ be an arbitrary real vector with $\|\hat{\boldsymbol{c}}_0\|=1$.

(2) Collect the measured data $u(t)$ and $y(t)$,form stacked vector $\boldsymbol{Y}(p,t)$ by Eq.(19) , $\boldsymbol{G}(t)$ by Eq.(5) the information vector $\hat{\boldsymbol{\varphi}}(t)$ by Eq.(23) and $\hat{\boldsymbol{\Phi}}_0(p,t)$ by Eq.(20).

(3) Compute and form $\boldsymbol{\Psi}(\hat{\boldsymbol{b}}_{t-1},t)$ by Eq. (21) and $\boldsymbol{\Phi}(\hat{\boldsymbol{c}}_{t-1},t)$ by Eq. (22).

(4) Form information matrix $\hat{\boldsymbol{\Xi}}(t)$ by Eq. (17) and compute innovation vector $\hat{\boldsymbol{E}}(p,t)$ by Eq. (18).

(5) Compute $\hat{\boldsymbol{\Pi}}_{23}(t)$ by Eq. (16) and $\hat{\boldsymbol{\Pi}}(t)$ by Eq. (15).

(6) Update the parameter estimation $\hat{\boldsymbol{\theta}}(t)$ by Eq. (13).

(7) Normalize $\hat{\boldsymbol{c}}_t$ by Eq. (24) and Eq. (25) with the first positive element.

(8) Increase t by 1 and go to Step 2.

3.2 Example

Consider the following Hammerstein nonlinear system:

$$A(z)y(t)=B(z)\overline{\mu}(t)+D(z)v(t)$$

$$A(z)=1+\alpha_1 z^{-1}+\alpha_2 z^{-2}=1-1.07z^{-1}+0.675z^{-2},\quad B(z)=\beta_1 z^{-1}+\beta_2 z^{-2}=1.55z^{-1}+1.20z^{-2},$$

$$D(z)=1+d_1z^{-1}=1+0.13z^{-1},$$

$$\overline{\mu}(t)=g(\mu(t))=c_1\mu(t)+c_2\mu^2(t)+c_3\eta^3(t)$$
$$=0.80\mu(t)+0.50\mu^2(t)+0.33166\mu^3(t),$$

$$\boldsymbol{\theta}=[\alpha_1,\alpha_2,d_1,\beta_1,\beta_2,c_1,c_2,c_3]^{\mathrm{T}}.$$

In simulation, the input $\{u(t)\}$ is taken as a persistent excitation signal sequence,the noise $\{v(t)\}$ is taken as a white noise sequence with zero mean and and variance $\sigma^2=0.30^2$, and the data length is taken as p=100 and p=160.Adopting the the Newton extended recursive algorithm(H-ENR) to estimate the paramelers of this Hammerstein-CARMA process,the corresponding noise-to-signal ratio is $\delta_{ns}=8.76\%$, where the noise-to-signal ratio δ_{ns} is as follows($h(t)$ and $x(t)$ in **Figure 1**):

$$\delta_{ns}=\sqrt{\frac{\operatorname{var}[h(t)]}{\operatorname{var}[x(t)]}}\times 100\%,$$

$$h(t)=\frac{D(z)}{A(z)}v(t),\qquad x(t)=\frac{B(z)}{A(z)}\overline{\mu}(t).$$

The parameter estimates and their errors are shown in **Table 1** , the estimation errors versus t are shown in **Figure 2.**

From **Table 1** and **Figure 2**, we can get the following conclusions:

(1) For the confirmed Hammerstein-CARMA model($v(t)$ =0 or σ^2=0),the Newton extended recursive algorithm can converge to the true value faster than the extended projection algorithm. For the stochastic Hammerstein-CARMA model ($\sigma^2\neq 0,\sigma^2=0.30^2$),the parameter estimation of Newton extended recursive algorithm fluctuates greatly, especially for the small stacked data length p, and its estimation errors cannot converge to zero even if the data length t tends to infinity. The reason is that the increment of the extended recursive algorithm does not approach zero. However, when the length of stacked data length p increases, the parameter estimation will become getting more stationary, as shown **Table 1** and **Figure 2** with p=100 and p=160.

(2) The parameter estimation errors rapidly converges to a small constant as the data length t increases. As the data length t goes to infinity, this constant is going to get very small and close to zero. This shows that the extended Newton recursive algorithm is effective.

4 Conclusions

This paper studies the parameter estimation methods for the nonlinear Hammerstein-CARMA model. A extended Newton recursive(H-ENR) algorithms are derived based on the Newton method. Aiming at the difficulty that the information vector of Hammerstein-CARMA model contains unmeasured noiseterms, the principle of recursion identification is applied. The unknown noise terms contained in the information vector are replaced by its estimated value, and the estimated value is calculated by the parameterestimated value of the previous time or the previous time. Compared with the extended stochastic gradient algorithms the H-ENR algorithm has improved parameter estimation accuracy.The numerical example shows that the parameter estimates for the proposed H-ENR algorithm converge to their true values.At present, there is a lot of work tobe done in the study of nonlinear systems. ln this paper, only single-input single-output nonlinearsystems are studied. How to extend it to multi-inputmulti-output nonlinearsystems and apply it in the field is the next problem to be considered.

Table 1 The H-ENR estimates and errors with $p=100$ and $p=160$ ($\sigma^2=0.30^2$)

p	t	α_1	α_2	d_1	β_1	β_2	c_1	c_2	c_3	$\delta(100\%)$
100	1	-1.06888	0.67413	3.04450	1.34687	0.90049	0.80049	0.48525	0.35177	115.55488
	2	-1.05271	0.65084	0.48409	1.50838	0.94563	0.80375	0.49613	0.32839	17.27314
	5	-1.08051	0.66901	-0.47730	1.48392	1.23121	0.79981	0.50128	0.33017	24.07139
	10	-1.07007	0.67512	0.05824	1.54652	1.19742	0.80015	0.49985	0.33151	2.82854
	15	-1.07022	0.67525	0.20194	1.54906	1.19976	0.80017	0.49991	0.33140	2.83091
	20	-1.07026	0.67529	0.15804	1.54935	1.19966	0.79982	0.50015	0.33187	1.10366
	50	-1.07013	0.67531	0.17327	1.55137	1.20127	0.80093	0.49948	0.33020	1.70550
160	1	-1.06684	0.67191	3.65162	1.34688	0.89635	0.80869	0.47924	0.34109	139.30408
	2	-1.05152	0.64952	0.51147	1.52817	0.94528	0.79505	0.50071	0.34233	18.11615
	5	-1.08051	0.66901	-0.47730	1.48392	1.23121	0.79981	0.50128	0.33017	24.07139
	10	-1.07007	0.67512	0.05824	1.54652	1.19742	0.80015	0.49985	0.33151	2.82854
	15	-1.07022	0.67525	0.20194	1.54906	1.19976	0.80017	0.49991	0.33140	2.83091
	20	-1.07026	0.67529	0.15804	1.54935	1.19966	0.79982	0.50015	0.33187	1.10366
	50	-1.07013	0.67531	0.17327	1.55137	1.20127	0.80093	0.49948	0.33020	1.70550
True values		-1.07000	0.67500	0.13000	1.55000	1.20000	0.80000	0.50000	0.33166	

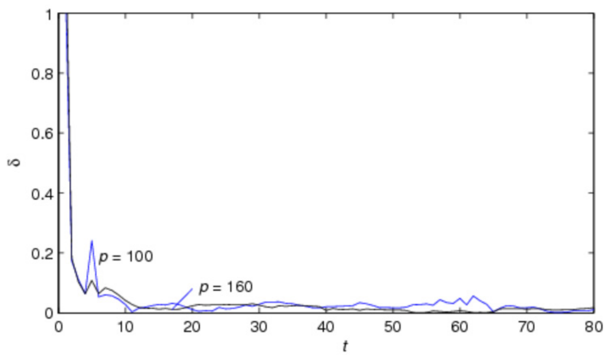

Figure 2 The H-ENR estimation errors t versus ($\sigma^2=0.30^2$)

5 Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities under Grant 31920210075.

6 References

[1] L. Ljung, System Identification: Theory for the User, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.

[2] F. Ding, System Identification - New Theory and Methods, Science Press, Beijing, 2013.

[3] Y.Cao, P. Li，Y. Zhang, Parallel processing algorithm for railway signal fault diagnosis data based on cloud computing，Future Gen. Comput. Syst. 88 (2018)279-283.

[4] Abd-Elrady, E.(2008). A recursive prediction error algorithm for digital predistortion of FIR Wiener systems. In Communication systems， networks and digital signal processing (CSNDSp08)(pp.698-701).

[5] L. Xu and F. Ding, "Parameter estimation for control systems based on impulse responses," International Journal of Control Automation and Systems, vol. 15, no. 6, pp. 2471-2479, December 2017.

[6] Lijuan Wan, Feng Ding, Ximei Liu, Chunping Chen. "A New Iterative Least Squares Parameter Estimation Approach for Equation-error Autoregressive Systems", International Journal of Control, Automation and Systems, volume 18, pages 780–790, November 2019 .

[7] Abd-Elrady, E. (2009). Adaptive predistortion of Wiener and Hammerstein systems using spectral magnitude matching. In Proceedings of the 11th IASTED conference on signal and image processing (SIP'09), Honolulu, Hawaii, USA (pp. 17-19).
[8] Golub G, Pereyra V. Separable nonlinear least squares: the variable projection method and its applications [J]. Inverse Problems, 2003, 19(2): R1-R26.
[9] I. Mansouri, A. Gholampour, O. Kisi, and T. Ozbakkaloglu, "Evaluation of peak and residual conditions of actively confined concrete using neuro-fuzzy and neural computing techniques," Neural Computing and Applications, vol. 29, no. 3, pp. 873-888, February 2018.
[10] Tulsyan A, Huang B, Gopaluni R B, Fraser F J. On simultaneous on-line state and parameter estimation in non-linear state-space models. Journal of Process Control, 2013,23(4):516-526.
[11] Y. B. Gong, S. X. Yang, H. L. Ma, and M. Ge, "Fuzzy regression model based on geometric coordinate points distance and application to performance evaluation," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 1, pp. 395-404, January 2018.
[12] J. J. Rubio, "SOFMLS: online self-organizing fuzzy modified least square network," IEEE Transactions on Fuzzy Systems, vol. 17, no. 6, pp. 1296-1309, December 2009.
[13] Y. P. Wang, L. C. Wang, D. H. Kong, and B. C. Yin, "Extrinsic least squares regression with closed-form solution on product grassmann manifold for video-based recognition," Mathematical Problems in Engineering, vol. 2018, no. 1, pp. 1-7, March 2018.
[14] Narendra K S, Gallman P G. An iterative method for the identification of nonlinear systems using a Hammerstein model [J]. IEEE Transactions on Automatic Control, 1996, 11(3):546-550.
[15] Stoica P. On the convergence of an iterative algorithm used for Hammerstein system identification [J]. IEEE Transactions on Automatic Control, 1981, 26(4):967-969.
[16] Rangan S, Wolodkin G, Poolla K. Identification methods for Hammerstein systems [C]. Proceedings of Control and Decision, New Orleans, 1995, 697-702.
[17] Chang F, Luus R. A noniterative method for identification using Hammerstein model [J]. IEEE Transactions on Automatic Control, 1971, 16(5):464-468.
[18] Cerone V, Regruto D. Parameter bounds for discrete-time Hammerstein models with bounded output errors [J]. IEEE Transaction on Automatic Control, 2003, 48(10):1855-1860.
[19] Bai E W, Li D. Convergence of the iterative Hammerstein system identification algorithm [J]. IEEE Transactions on Automatic Control, 2004, 49(11):1929-1940.
[20] Ding F, Chen T. Identification of Hammerstein nonlinear ARMAX systems [J]. Automatica, 2005, 41(9):1479-1489.
[21] Ding F, Shi Y, Chen T. Gradient-based identification methods for Hammerstein nonlinear ARMAX models [J].Nonlinear Dynamics, 2006, 45(1-2):31-43.
[22] Vanbeylen L, Pintelon R, Schoukens J. Blind maximum likelihood identification of Hammerstein systems [J]. Automatica, 2008, 44(12):3139-3146.
[23] Giri F, Rochdi Y, Chaoui F.Z, Brouri A. Identification of Hammerstein systems in presence of hysteresis-backlash and hysteresis-relay nonlinearities [J]. Automatica, 2008, 44(3):767-775.
[24] Jing Dahai, Liu Xiao-Ping. Online Identification Method for Nonlinear Models[J]. Control Engineering of China, 2007, 14(5):482-484.
[25] Yuan Xiaolei, Bai Yan, Dong Ling. Nonlinear System Identification Based on Genetic Programming [J]. Control Engineering of China, 2009, 16(1):52-55.
[26] Liu Y. Bai E W. Iterative identification of Hammerstein systems [J]. Automatica, 2007, 43(2):346-354.
[27] Zhang Xueqing, Wang Xiangfeng, Jiang Jie, Wu Zhaoming, Wang Fang. Research of Manipulator Vibration Controlling Based on Fuzzy Identification [J]. Control Engineering of China, 2007,14(s1):57-59.
[28] Feng Ding, Peter Xiaoping Liu, Guangjun Liu. ldentification methods for Hammerstein nonlinear systems. Digit. Signal Process.21(2):215-238(2011) .
[29] Liu, Y.,& Bai,E.W. (2007). Iterative identification of Hammerstein systems. Automatica, 43(2),346-354.