Research Article

Instrumental Variable-Based OMP Identification Algorithm for Hammerstein Systems

Shuo Zhang, 1 Dongqing Wang, 1,2 and Yaru Yan 1

1College of Automation Engineering, Qingdao University, Qingdao 266071, China
2Collaborative Innovation Center for Eco-Textiles of Shandong Province, Qingdao 266071, China

Correspondence should be addressed to Dongqing Wang; dqwang64@163.com

Received 31 March 2018; Accepted 6 May 2018; Published 22 July 2018

Academic Editor: Jing Na

Copyright © 2018 Shuo Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hammerstein systems are formed by a static nonlinear block followed by a dynamic linear block. To solve the parameterizing difficulty caused by parameter coupling between the nonlinear part and the linear part in a Hammerstein system, an instrumental variable method is studied to parameterize the Hammerstein system. To achieve in simultaneously identifying parameters and orders of the Hammerstein system and to promote the computational efficiency of the identification algorithm, a sparsity-seeking orthogonal matching pursuit (OMP) optimization method of compressive sensing is extended to identify parameters and orders of the Hammerstein system. The idea is, by the filtering technique and the instrumental variable method, to transform the Hammerstein system into a simple form with a separated nonlinear expression and to parameterize the system into an autoregressive model, then to perform an instrumental variable-based orthogonal matching pursuit (IV-OMP) identification method for the Hammerstein system. Simulation results illustrate that the investigated method is effective and has advantages of simplicity and efficiency.

1. Introduction

Nonlinear system modeling and identification are very important in theory and application [1–6]; block-oriented nonlinear systems, which combine nonlinear and linear blocks in various styles, are the typical representation of nonlinear systems. Hammerstein systems, a static nonlinear block plus a dynamic linear block, are a type of commonly used block-oriented nonlinear systems, and a lot of work has contributed in identification methods for Hammerstein systems. The traditional identification methods for Hammerstein systems mainly include the overparameterization model-based methods [7–10] and the recursive/iterative identification methods [11–14]. The highly efficient identification methods include the key-term separation principle-based identification methods [15–17], the hierarchical identification methods [18, 19], the filtering technique based-identification methods [20–22], the maximum likelihood estimation methods [23–25], and the evolution optimization methods [26–28]. Recently, Chen et al. investigated a particle swarm optimization method [29], Krishnanathan et al. discussed a continuous-time nonlinear systems using approximate Bayesian computation [30], and Wang et al. studied a model recovery for Hammerstein systems using the auxiliary model-based orthogonal matching pursuit method [31] and so on.

It is difficult to parameterize Hammerstein systems due to existing parameter coupling between the nonlinear part and the linear part of Hammerstein systems. In order to solve this problem, an instrumental variable-based method is studied to parameterize the Hammerstein systems. Further, to achieve in simultaneously identifying parameters and orders and to promote the computational efficiency of the estimated method, a sparsity-seeking orthogonal matching pursuit optimization method of compressive sensing is extended to identify parameters and orders of the Hammerstein systems. The idea is, by the filtering technique and the instrumental variable method, to transform the Hammerstein system into a simple form with a separated nonlinear expression, so as to easily parameterize the system into...
autoregressive form, and to perform the instrumental variable-based orthogonal matching pursuit (IV-OMP) identification method for the Hammerstein systems.

In the compressive sensing theory [32–34], there mainly exist two sparse signal recovery methods: the orthogonal matching pursuit algorithms and the basis pursuit algorithms. The orthogonal matching pursuit (OMP) algorithm is a kind of greedy parameter recovering method [35–37], it selects the best fitting column of the measurement matrix and the corresponding sparse signal in each selected step. Due to the selection being orthogonal, the OMP algorithm has a lower computational complexity compared with the basis pursuit algorithms [38–40].

Recently, Mao et al. investigated parameter estimation algorithms for Hammerstein time-delay systems based on the OMP scheme to estimate the system parameters and the time delay [41] by means of the compressed sensing recovery theory and the auxiliary model identification idea. In this paper, an instrumental variable-based orthogonal matching pursuit (IV-OMP) algorithm is investigated to simultaneously estimate the orders and parameters of a Hammerstein system. The contributions lie in four aspects:

(i) To solve the parameterizing difficulty caused by the coupling between the nonlinear part and the linear part in Hammerstein systems, a filtering technique-based instrumental variable method is studied to parameterize the Hammerstein systems.

(ii) To achieve in simultaneously identifying parameters and orders and to promote the computational efficiency of the system, an instrumental variable-based orthogonal matching pursuit (IV-OMP) optimization method of compressive sensing is extended to identify parameters and orders of the systems.

(iii) The advantage of the proposed IV-OMP method over the traditional methods is that it is not necessary to collect a lot of data and invest a lot of power into the parameter identification based on the sparse principle.

(iv) Simulation results illustrate that the investigated method is effective and has advantages of simplicity and efficiency.

The rest of the paper is organized as follows. Section 2 demonstrates the identification problem of a Hammerstein system. Section 3 studies the parameterizing method of the Hammerstein system. Section 4 presents the IV-OMP identification algorithm by using the instrumental variables. Section 5 provides a numerical example for the proposed method. Finally, the concluding remarks are involved in Section 6.

### 2. The Problem Formulation

For the narrative convenience, we define some notation. "M = N" represents "M is defined as N"; z represents a unit forward shift operator: zy(t) = y(t + 1) and z−1y(t) = y(t − 1);

\[ x(t) = f[u(t)] = \sum_{k=1}^{n_c} c_k f_k[u(t)], \]  
\[ A(z)y(t) = B(z)x(t) + D(z)\nu(t), \]

where u(t) and y(t) are the system input and output, x(t) is an internal variable, and \( \nu(t) \) is stochastic white noise with zero mean; the input nonlinearity f is modeled as a linear combination of basis functions \( f_k \); \( n_c \) is the number of the basis functions; the linear block is a CARMA model; A(z), B(z), and D(z) are polynomials in the unit backward shift operator \( z^{-1} \)

Figure 1: A system described by the Hammerstein CARMA model.

\[ \ddot{x}(t) \text{ stand for the estimate of } x \text{ at time } t. \]  

\[ x(t) = f[u(t)] = \sum_{k=1}^{n_c} c_k f_k[u(t)], \]

where u(t) and y(t) are the system input and output, x(t) is an internal variable, and \( \nu(t) \) is stochastic white noise with zero mean; the input nonlinearity f is modeled as a linear combination of basis functions \( f_k \); \( n_c \) is the number of the basis functions; the linear block is a CARMA model; A(z), B(z), and D(z) are polynomials in the unit backward shift operator \( z^{-1} \)

\[ A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a}, \]
\[ B(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b}, \]
\[ D(z) = 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_{n_d} z^{-n_d}. \]

If the order \( n_a \) is known and the orders \( n_b, n_c, \) and \( n_d \) are unknown, then we face two parameterizing problems for the Hammerstein system:

(i) The coupling between the nonlinear part and the linear part in a Hammerstein system makes the parameterization of the Hammerstein system more difficult.

(ii) Simultaneous and efficient identification of system parameters and orders is difficult to be achieved, due to existing unknown orders.

### 3. The Parametrization of The Hammerstein CARMA System

Divide both sides of (2) by the filtering function B(z), we get

\[ \frac{A(z)}{B(z)} y(t) = x(t) + \frac{D(z)}{B(z)} \nu(t). \]

Define the instrumental variables z(t) and w(t) as

\[ z(t) = \frac{A(z)}{B(z)} y(t), \]
\[ w(t) = \frac{D(z)}{B(z)} \nu(t). \]

Then, (4) can be written as
\[ z(t) = x(t) + w(t). \] (7)

Replacing the expressions of \( A(z) \) and \( B(z) \) and \( D(z) \) into (5) and (6), respectively, we can get

\[ z(t) = -\sum_{i=1}^{n_b} b_i z(t-i) + y(t) + \sum_{i=1}^{n_a} a_i y(t-i), \] (8)

\[ w(t) = -\sum_{i=1}^{n_b} b_i w(t-i) + v(t) + \sum_{i=1}^{n_d} d_i v(t-i). \] (9)

Substituting \( z(t) \) and \( w(t) \) in (8) and (9) and \( x(t) \) in (1) into (7) gives

\[
y(t) = -\sum_{i=1}^{n_b} a_i y(t-i) + \sum_{i=1}^{n_c} c_i f_j[u(t)] \\
+ \sum_{i=1}^{n_b} b_i(z(t-i) - w(t-i)) \\
+ \sum_{i=1}^{n_d} d_i v(t-i) + v(t). \]

Since the orders \( n_b, n_a, \) and \( n_d \) are unknown, we assume a sufficient length of order \( L \) for \( n_b, n_a, \) and \( n_d(L > n_b, L > n_a, L > n_d). \) Then, (10) can be rewritten as

\[
y(t) = -\sum_{i=1}^{n_b} a_i y(t-i) + \sum_{i=1}^{L} c_i f_j[u(t)] \\
+ \sum_{i=1}^{n_b} b_i(z(t-i) - w(t-i)) \\
+ \sum_{i=1}^{n_d} d_i v(t-i) + v(t). \]

Define the information vectors \( \varphi(t), \varphi_a(t), \) and \( \varphi_v(t) \) as

\[
\varphi(t) = \begin{bmatrix} \varphi_a(t) \\ \varphi_v(t) \end{bmatrix} \in \mathbb{R}^N, \quad N = 3L + n_a,
\]

\[
\varphi_a(t) = [y(t-1), -y(t-2), \ldots, -y(t-n_a), f_1[u(t)], f_2[u(t)], \ldots, f_L[u(t)]]^T \in \mathbb{R}^{L+n_a},
\]

\[
\varphi_v(t) = [z(t-1) - w(t-1), z(t-2) - w(t-2), \ldots, z(t-L) - w(t-L), v(t-1), \ldots, v(t-2), \ldots, v(t-L)]^T \in \mathbb{R}^{2L},
\]

then (15) can be described as

\[
Y = \Phi \Theta + V. \] (17)

If there are enough measurements, that is, \( m \) reaches several thousands, we can get the least squares estimate of \( \Theta \) as follows:

\[
\hat{\Theta}_{LS} = (\Phi^T \Phi)^{-1} \Phi^T Y. \] (18)

But with an assumed big order length \( L \geq 5 \) and \( N, \) the above least squares algorithm leads a big computational burden and is not suitable for solving system parameters and orders on line. To avoid this problem and get an efficient
identification algorithm, we choose a more advantageous method over the traditional least squares method; the aim is to identify the parameter vector \( \Theta \) with less observations \( K < m < n \) by using the OMP theory based on the mentioned instrumental variable method.

According to the CS theory, we can regard the parameter vector \( \Theta \) as a sparse signal. Let \( \| \Theta \|_0 = K \) be the number of the nonzero entries in \( \Theta \), that is, the sparsity level of \( \Theta \), then the identification problem can be described as an orthogonal matching pursuit (OMP) method:

\[
\hat{\Theta} = \arg \min \| \Theta \|_0, \quad \text{s.t.} \quad \| y - \Phi^T \Theta \|_2 < \varepsilon,
\]

where \( \hat{\Theta} \) is the estimate of \( \Theta \) and \( \varepsilon (\varepsilon > 0) \) is the error tolerance.

### 4. The IV-OMP Identification Algorithm

For the sparse parameter vector \( \Theta \), if \( K = \| \Theta \|_0 (K < N) \), there will be \( K \) nonzero scalar components in \( \Theta \). What we should do in this identification method is to pick out and recover valid data in \( \Theta \) from \( \Phi \) by using the OMP theory.

Define \( \phi_i \), as the \( i \)-th column vector of \( \Phi \) (a \( \phi_i \) is also called an atom), and \( \theta_i \) as the \( i \)-th element of \( \Theta(i = 1, 2, \ldots, N) \), that is,

\[
\Phi = [\phi_1, \phi_2, \ldots, \phi_N], \quad \Theta = [\theta_1, \theta_2, \ldots, \theta_N]^T.
\]

Then, the vector \( Y \) in (17) can be written as

\[
Y = \sum_{i=1}^{N} \phi_i \theta_i + V.
\]

Obviously, the output vector \( Y \) contains a linear combination of all atoms \( \phi_i (i = 1, 2, \ldots, N) \). The main idea is to find the \( K \) nonzero items and corresponding valid atoms at the right-hand side of (21).

To describe the IV-OMP recovery algorithm, we need to give some notions. Let \( k = 1, 2, \ldots \) be the iterative number and \( \lambda_k \) be the index of the solution support \( \phi_{\lambda_k} \) in \( \Phi \) at the \( k \)-th iteration; \( \Lambda_k \) is a set composed of \( \lambda_i \), \( i = 1, 2, \ldots, k \); \( \Phi_{\Lambda_k} \) is the submeasurement matrix composed by the \( k \) columns of \( \Phi \) indexed by \( \Lambda_k \); \( r_k \) denotes the residual at the \( k \)-th iteration; and \( \hat{\Theta}_k \) is the parameter estimation at the \( k \)-th iteration. The algorithm is initialized as \( r_0 = Y, \Lambda_0 = \emptyset \) and \( \hat{\Theta}_{\Lambda_0} = 0 \).

Define a cost function at the \( k \)-th iteration

\[
J(\theta_i) = \| r_{k-1} - \phi_i \theta_i \|_2^2, \quad i = 1, 2, \ldots, N.
\]

Remark 1. Minimizing \( J(\theta_i) \) with respect to \( \theta_i \) means that the derivation of \( J(\theta_i) \) with respect to \( \theta_i \) is equal to 0, that is,

\[
\frac{\partial J(\theta_i)}{\partial \theta_i} = -2\phi_i^T r_{k-1} - \phi_i \theta_i = 0,
\]

and we get

\[
\theta_i = \frac{\phi_i^T r_{k-1}}{\| \phi_i \|^2}_T.
\]

Substitute the above \( \theta_i \) into (22) to get a minimized \( J(\theta_i) \) as

\[
J(\theta_i)_{\text{min}} = \left( \frac{\phi_i^T r_{k-1}}{\| \phi_i \|^2} - r_{k-1} \right)^T \left( \frac{\phi_i^T r_{k-1}}{\| \phi_i \|^2} - r_{k-1} \right)
\]

\[
= \left( \frac{r_{k-1}^T \phi_i^T - r_{k-1}}{\| \phi_i \|^2} \right)^T \left( \frac{r_{k-1}^T \phi_i^T - r_{k-1}}{\| \phi_i \|^2} \right)
\]

\[
= \frac{r_{k-1}^T \phi_i^T \phi_i r_{k-1}}{\| \phi_i \|^2} - \frac{r_{k-1}^T \phi_i^T r_{k-1}}{\| \phi_i \|^2} + \frac{1}{\| \phi_i \|^2}.
\]

Remark 2. The result \( J(\theta_i)_{\text{min}} = \| r_{k-1} \|^2 - (\phi_i^T r_{k-1}/\| \phi_i \|^2)^2 \) says that minimizing \( J(\theta_i) \) is equivalent to the quest for the largest inner product between the residual \( r_{k-1} \) and the normalized column vector \( \phi_i \) of \( \Phi \).

The next step is to find the index \( i \) and the column \( \phi_i \) corresponding to the largest inner product between the residual \( r_{k-1} \) and the normalized column vectors \( \phi_i \), and assign them into \( \lambda_k \) and \( \phi_{\lambda_k} \) at the \( k \)-th step:

\[
\lambda_k = \arg \max_{i=1,2,\ldots,N} \left| \frac{r_{k-1}^T \phi_i}{\| \phi_i \|^2} \right|.
\]

Remark 3. Minimizing the least squares cost function \( J(\Theta_{\lambda_k}) = \| Y - \Phi_{\lambda_k} \Theta_{\lambda_k} \|^2 \) is equal to taking the derivation of \( J(\Theta_{\lambda_k}) \) with respect to \( \Theta_{\lambda_k} \),

\[
\frac{\partial J(\Theta_{\lambda_k})}{\partial \Theta_{\lambda_k}} = \Phi_{\lambda_k}^T (Y - \Phi_{\lambda_k} \Theta_{\lambda_k}) = \Phi_{\lambda_k}^T r_k = 0.
\]

It means that the residual \( r_k \) is orthogonal to the subinformation matrix \( \Phi_{\lambda_k} \). Therefore, the index corresponding to the largest inner product between the residual \( r_{k-1} \) and the normalized column vector \( \phi_i \) is computed by

\[
\lambda_k = \arg \max_{i \in S} \left| \frac{r_{k-1}^T \phi_i}{\| \phi_i \|^2} \right|, \quad S = \{1, 2, \ldots, N\}.
\]

Update the support set \( \Lambda_k \) and the subinformation matrix \( \Phi_{\Lambda_k} \) by

\[
\Lambda_k = [\Lambda_{k-1}, \lambda_k], \quad \Phi_{\Lambda_k} = \left[ \Phi_{\Lambda_{k-1}}, \phi_{\lambda_k} \right].
\]
With the obtained index set $\Lambda_k$ and the corresponding subinformation matrix $\Phi_{\Lambda_k}$, the least squares estimation for the nonzero parameters at the $k$th iteration can be achieved. Define a cost function:

$$J(\Theta_{\Lambda_k}) = ||Y - \Phi_{\Lambda_k} \Theta_{\Lambda_k}||^2. \quad (30)$$

Minimizing $J(\Theta_{\Lambda_k})$ should get the least squares estimates of the nonzero parameters. But there exist the unknown instrumental variables $z(t-i)$ and $w(t-i)$ and the unmeasurable noise term $v(t-i)$ in the information vector $\Phi(t)$ of $\Theta$; the parameter vector $\Theta_{\Lambda_k}$ cannot be estimated by using the standard OMP methods in the CS theory. The method is to use their estimates $\hat{z}(t-i)$, $\hat{w}(t-i)$, and $\hat{v}(t-i)$ to replace them; the computation of the estimates $\hat{z}(t-i)$, $\hat{w}(t-i)$, and $\hat{v}(t-i)$ is as follows:

(i) Expand the orders $n_0$, $n_c$, and $n_d$ in (1), (2), (8), and (9).

(ii) $\hat{z}(t-i)$ is computed by replacing $a_i$ and $b_i$ with their estimates $\hat{a}_{ik}$ and $\hat{b}_{ik}$ in (8).

(iii) $\hat{v}(t-i)$ is computed by replacing $a_i$, $b_i$, $c_i$, and $d_i$ with their estimates $\hat{a}_{ik}$, $\hat{b}_{ik}$, $\hat{c}_{ik}$, and $\hat{d}_{ik}$ in (2) with (1) being inserted.

(iv) $\hat{w}(t-i)$ is computed by replacing $b_i$ and $d_i$ with their estimates $\hat{b}_{ik}$ and $\hat{d}_{ik}$ in (9). The results are

$$\hat{z}(t) = -\sum_{i=1}^{L} \hat{b}_{ik} \hat{z}(t-i) + \sum_{i=1}^{n_a} \hat{a}_{ik} y(t-1) + y(t),$$

$$\hat{v}(t) = \sum_{i=0}^{n_c} \hat{a}_{ik} y(t-i) - \sum_{i=0}^{L} \hat{b}_{ik} \sum_{i=1}^{L} \hat{c}_{ik} f_k[u(t-i)]$$

$$- \sum_{i=1}^{L} \hat{d}_{ik} \hat{v}(t-i), \quad \hat{b}_{i,0} = 1$$

$$\hat{w}(t) = -\sum_{i=1}^{L} \hat{b}_{ik} \hat{w}(t-i) + \sum_{i=1}^{L} \hat{d}_{ik} \hat{v}(t-i) + \nu(t), \quad \quad (31)$$

and the estimates of $\Phi_v(t)$, $\Phi(t)$, and $\Phi$ at iteration $k$ can be written as

$$\hat{\Phi}_{vk}(t) = \left[ \hat{z}(t-1) - \hat{w}(t-1), \hat{z}(t-2) - \hat{w}(t-2), \ldots, \hat{z}(t-L) - \hat{w}(t-L), \hat{v}(t-1), \hat{v}(t-2), \ldots, \hat{v}(t-L) \right]^T \in \mathbb{R}^{2L},$$

$$\hat{\Phi}(t) = \left[ \hat{\Phi}_v(t), \hat{\Phi}(t) \right].$$

With replacing the above unknown variables with their estimates, the least squares estimate $\hat{\Theta}_{\Lambda_k}$ for the parameter vector $\Theta_{\Lambda_k}$ at the $k$th step is given by

$$\hat{\Theta}_{\Lambda_k} = \left( \hat{\Phi}_{\Lambda_k}^T \hat{\Phi}_{\Lambda_k} \right)^{-1} \hat{\Phi}_{\Lambda_k}^T Y. \quad (33)$$

**Remark 4.** At the beginning, the IV-OMP algorithm with the estimates $\hat{z}(t-1)$, $\hat{w}(t-1)$, and $\hat{v}(t-1)$ in the subinformation matrix $\Phi_{\Lambda_k}$ causes an inaccurate support atom selecting. With the iteration $k$ increasing, these estimated variables become more accurate, the misselected support atoms certainly unmeet the threshold requirement, and the corresponding element in the estimated parameter support set will be a small nonzero value. Thus, we set an appropriate small threshold $\epsilon$ to filter the parameter estimate $\hat{\Theta}_{\Lambda_k}$. If $|\hat{\theta}_{hk}| < \epsilon$ (where $\hat{\theta}_{hk}$ is the $h$th element of $\hat{\Theta}_{\Lambda_k}$), eliminate $\hat{\theta}_{hk}$ from $\hat{\Theta}_{\Lambda_k}$ and the corresponding $\hat{\phi}_{hk}$ from $\hat{\Phi}_{\Lambda_k}$.

$$\hat{\Theta}_{\Lambda_k} = \left[ \hat{\Theta}_{\Lambda_k} \setminus \hat{\theta}_{hk}, \hat{\Phi}_{\Lambda_k} \right] \quad (34)$$

Because $\hat{\theta}_{hk}$ is the $h$th element in $\hat{\Theta}_{\Lambda_k}$ and $\hat{\phi}_{hk}$ is the $h$th column in $\hat{\Phi}_{\Lambda_k}$, the expressions of eliminating $\hat{\theta}_{hk}$ from $\hat{\Theta}_{\Lambda_k}$ and $\hat{\phi}_{hk}$ from $\hat{\Phi}_{\Lambda_k}$ in MATLAB are

$$\hat{\Theta}_{\Lambda_k} : \hat{\Theta}_{\Lambda_k}(h) = \emptyset,$$

$$\hat{\Phi}_{\Lambda_k} : \hat{\Phi}_{\Lambda_k}(h) = \emptyset. \quad (35)$$

Then we have

$$r_k = Y - \hat{\Phi}_{\Lambda_k} \hat{\Theta}_{\Lambda_k}. \quad (36)$$

Briefly, the proposed IV-OMP algorithm can be implemented as in Algorithm 1.
5. Examples

Consider the following Hammerstein CARMA system

\[
\begin{align*}
A(z)y(t) &= B(z)x(t) + D(z)v(t), \\
x(t) &= -1.70u(t) + 0.80u^2(t), \\
A(z) &= 1 + 1.20z^{-1} + 0.60z^{-2}, \\
B(z) &= 1 + 0.12z^{-1} - 0.06z^{-2}, \\
D(z) &= 1 + 0.11z^{-1} - 0.10z^{-2}, \quad (37)
\end{align*}
\]

\[
\Theta = \left[ \begin{array}{c}
1.20, 0.60, -1.70, 0.80, 0, 0, 0, 0, 0.12, -0.06, 0, 0, 0, 0.11, -0.10, 0, 0, 0, 0, 0, 0
\end{array} \right]^T
\]
Table 1: the recursive parameter estimates and errors by using different algorithms.

| Algorithm | \( k \) | \( \hat{a}_{1,k} \) | \( \hat{a}_{2,k} \) | \( \hat{c}_{1,k} \) | \( \hat{c}_{2,k} \) | \( \hat{c}_{3,k} \) | \( \hat{c}_{4,k} \) | \( \hat{b}_{1,k} \) | \( \hat{b}_{2,k} \) | \( \hat{b}_{3,k} \) | \( \hat{b}_{4,k} \) | \( \hat{b}_{5,k} \) | \( \hat{d}_{1,k} \) | \( \hat{d}_{2,k} \) | \( \hat{d}_{3,k} \) | \( \hat{d}_{4,k} \) | \( \delta \) (%) |
|-----------|--------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|           | 1      | 0.7717          | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 87.52           |
|           | 2      | 0.7100          | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 88.71           |
|           | 3      | 0.824           | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0.5002          | 90.55           |
|           | 4      | 1.304           | 0.7615          | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 82.65           |
|           | 5      | 1.331           | 0.8617          | 0               | 0               | 0               | 0.2315          | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | -0.0846         | 83.87           |
|           | 6      | 1.399           | 0.8219          | 0               | 0               | 0               | 0.1992          | 0               | -0.2422         | 0               | 0               | 0.3025          | 0               | 0               | 0               | 0               | 0               | -0.2355         | 85.66           |
|           | 7      | 1.086           | 0.5736          | -1.857          | 0               | 0               | 0.3747          | 0.0527          | 0               | 0               | 0               | 0               | 0               | 0.0982          | 0               | 0               | 0               | 0.0080          | 40.00           |
| IV-OMP    | 8      | 1.126           | 0.5786          | -1.668          | 0.8567          | 0               | -0.0220         | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0.0314          | 0               | 0               | 0.0172          | 9.46            |
|           | 9      | 1.124           | 0.5833          | -1.693          | 0.8531          | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0.0362          | 0               | 0               | 0               | 0.0126          | 9.27            |
|           | 10     | 1.124           | 0.5855          | -1.686          | 0.8181          | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0.0370          | 0               | 0               | 0               | 0.0104          | 9.02            |
|           | 11     | 1.125           | 0.5732          | -1.697          | 0.8275          | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0               | 0.0531          | 0               | 0               | 0               | -0.0238         | 8.58            |
|           | 12     | 1.122           | 0.5754          | -1.690          | 0.8295          | 0               | 0               | 0               | 0               | 0               | 0               | 0.0303          | 0               | 0               | 0               | 0               | 0               | -0.0477         | 0.0038          | 7.54            |
|           | 13     | 1.163           | 0.5926          | -1.688          | 0.8159          | 0               | 0               | 0               | 0               | 0               | 0               | 0.0864          | -0.0610         | 0               | 0               | 0.0593          | -0.0686         | -0.0090         | 0               | 3.48            |
|           | 14     | 1.187           | 0.5913          | -1.682          | 0.8096          | 0               | 0               | 0               | 0               | 0               | 0.0976          | -0.0680         | 0               | 0               | 0.0858          | -0.0756         | -0.0016         | 0               | 2.18            |
|           | 15     | 1.196           | 0.5973          | -1.659          | 0.8035          | 0               | 0               | 0               | 0               | 0               | 0.1014          | -0.0626         | 0               | 0               | 0.0929          | -0.0726         | 0               | 0               | 2.46            |
| True values| 1      | 1.20            | 0.60            | -1.70           | 0.80            | 0               | 0               | 0               | 0.12            | -0.06           | 0               | 0               | 0.11            | -0.10           | 0               | 0               |
In the simulation, the input \( \{u(t)\} \) is taken as an uncorrelated persistently excited signal vector sequence with zero mean and unit variance, and \( \{v(t)\} \) is taken as a white noise sequence with zero mean and variance \( \sigma^2 = 0.10^2 \). Applying the proposed IV-OMP algorithm to estimate the parameters and the orders of this system, the parameter estimates and their errors are shown in Table 1 and Figures 2 and 3. The relative error of the parameters is

\[
\delta = \frac{\|\hat{\Theta}_k - \Theta\|}{\|\Theta\|} \times 100\%.
\] (38)

From Table 1 and Figures 2–3, we can get the following:

1. The parameter estimation errors become (generally) smaller and smaller with the iteration \( k \) increasing.
2. There exist 3 wrongly selected atoms corresponding to parameters \( \hat{c}_{4,k}, \hat{c}_{5,k}, \) and \( \hat{d}_{3,k} \) (black-colored lines in Figure 3). With the iteration \( k \) increasing, the wrongly chosen atoms are deleted, and the parameter estimation errors become smaller.

6. Conclusions

This paper is aimed at exploring the identification of both the parameters and orders of the Hammerstein CARMA system with limited sampling data. To solve the parameterizing difficulty caused by parameter coupling between the nonlinear part and the linear part in a Hammerstein system, firstly, by filtering the equation of the linear block with the coefficient function of the controlled term, we separate the parameter coupling between the linear block and the nonlinear block. Moreover, by using two instrumental variables, the Hammerstein system is parameterized into an autoregressive form. To achieve in simultaneously identifying parameters and orders and to promote the computational efficiency of the identification algorithm, an instrumental variable-based orthogonal matching pursuit (IV-OMP) optimization method of compressive sensing is extended to identify parameters and orders of the Hammerstein system. Simulation results illustrate that the investigated method is effective and has advantages of simplicity and efficiency. The proposed IV-OMP optimization method can be extended to the colored noise systems, the networked dynamic systems [42–46], and so on.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (no. 61573205) and the Shandong Provincial Natural Science Foundation of China (no. ZR2015FM017).

References


