IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 17, 2019, accepted February 27, 2019, date of publication March 21, 2019, date of current version April 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904096

Sliding Window Iterative Identification of Systems With Asymmetric Preload Nonlinearity Based on the Key Term Separation

JUNXIA MA, QIULIN FEI, AND WEILI XIONG

Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China

Corresponding author: Junxia Ma (jxma@jiangnan.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61803183, in part by the Fundamental Research Funds for the Central Universities under Grant JUSRP11923, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180591 and Grant BK20170198.

ABSTRACT The parameter estimation problem for the Hammerstein systems with asymmetric preload nonlinearity is considered in this paper. The nonlinearity is described by the piecewise function, which brings difficulty to the identification. By introducing a switching function, the static nonlinearity is described by an expression with unknown preload points and slopes. By using the key term separation technique, the unknown parameters from the nonlinear block and linear block are decoupled and collected in a parameter vector. Under the gradient iterative (GI) algorithm with finite measured data, the estimates of unknown parameters are obtained. Furthermore, combining the ideas of the recursive and iterative algorithms, a dynamic sliding window is designed. By updating the training data, the sliding window removes the oldest data and adds the newest sampled data to keep the length of the training data unchanged. The sliding window gradient-based iterative algorithm is proposed to estimate the unknown parameters. Moreover, compared with the stochastic gradient algorithm, the sliding window GI algorithm can improve the accuracy of parameter estimation and the utilization of the system data. The numerical simulation example is employed to validate the effectiveness of the proposed algorithms.

INDEX TERMS Iterative identification, sliding window, asymmetric preload nonlinearity, parameter estimation, Hammerstein model.

I. INTRODUCTION

System identification aims to determine the mathematical model describing the behavior of the system based on the time function of the input and output [1], [2]. The purpose of establishing a mathematical model by identification is to estimate the important parameters that characterize the behavior of the system, to predict the future evolution of the system output, and to design the controller [3]–[6]. The identification methods for linear systems has been greatly developed, and some classical parameter identification algorithms have emerged, such as the least squares method [7], [8] and the maximum likelihood method [9]. However, the phenomena of nonlinearities widely exist in engineering practice [10]–[13].

The associate editor coordinating the review of this manuscript and approving it for publication was Ludovico Minati.

Because of the complexity and diversity of nonlinear systems, there is no uniform method for identifying nonlinear models currently [14]–[16]. Different model structures are used to describe the characteristics of the systems for different nonlinearities. Hammerstein model [17], Wiener model [18] and Hammerstein-Wiener [19] are usually applied to fit the input nonlinear, output nonlinear and input-output nonlinear systems, respectively.

The Hammerstein model which consists of the interaction of a linear time-invariant dynamic subsystem and static nonlinear element can be used to capture some nonlinear behaviors presented in system input [20]. The identification of input nonlinear systems has received a lot of attention. For example, by using the overparametrization method, an optimal two-stage identification algorithm for a class of nonlinear systems was derived [21]. Other identification methods for a

class of nonlinear systems (bilinear systems) can be found in [22] and [23]. One case is that the nonlinearity of the Hammerstein model is described by a continuous nonlinear function, such as a polynomial about the input signal

$$h_t = \lambda_1 h_1(u_t) + \lambda_2 h_2(u_t) + \cdots + \lambda_n h_n(u_t),$$

where λ_i's are the unknown coefficients, $h_i(\cdot)$'s are the known basis function about the input, such as power function. Usually, when the nonlinearity is static and analytic, a polynomial which linear in the unknown parameters is a good choose.

However, hard input nonlinearities are common in real practice, such as saturation of the amplifier, preload system of the elevator. The non-smooth hard nonlinearity may not be approximated by a polynomial. The expressions of the nonlinearity can be a piecewise linear function or multi-segment nonlinearities, which are not linear in the unknown parameters. To obtain the parameter estimation, some approaches were discussed. For example, Bai [24] proposed a deterministic separable least squares identification algorithm for systems with hard input nonlinearities parameterized by a one-dimensional parameter, but the method is designed for the symmetric nonlinearity with a single parameter. Vörös studied an iterative parameter estimation algorithm for a Hammerstein system which is consisted of a two-segment nonlinearity followed by a linear dynamic system [25].

The iterative algorithm processes a batch of sampled data and makes efficient use of the measurement data at each iteration step. Hence, the iterative algorithm are usually applied for the optimal design, iterative learning and parameter estimation [26], [27]. However, the iterative algorithm needs to collect a batch of sampled data first and is implemented offline. Different from the iterative algorithm mentioned above, and combining the advantages of the recursive algorithm, the paper aims to design a dynamic sliding window to update the collected training data to identify a Hammerstein system with asymmetric preload nonlinearity. The main contributions are as follows.

- By defining a symbolic function, the asymmetric preload nonlinearity is described by an expression concluding all unknown preload points and slopes. Furthermore, using the key term separation technique, the unknown parameters from nonlinear block and linear block are decoupled and included in one parameter vector. The output equation is rewritten as a linear form in parameters.
- According to the negative gradient search principle, the stochastic gradient (SG) and the gradient iterative (GI) algorithm are derived for the Hammerstein systems by using the interactive identification technique.
- By design a dynamic sliding window to update the training data, a sliding window gradient based iterative algorithm is derived. The proposed algorithm can always use the latest batch of sampled data and maintains a high data utilization.

The remainder of this paper is organized as follows. Section II deduces the identification model for Hammerstein systems with asymmetric preload in detail. Section III presents the SG and GI identification algorithm for comparison. By using a sliding data window, Section IV develops a sliding window iterative identification algorithm based on the key term separation technique. Section V provides a numerical example to validate the proposed algorithms. Finally, some concluding remarks are presented in Section VI.

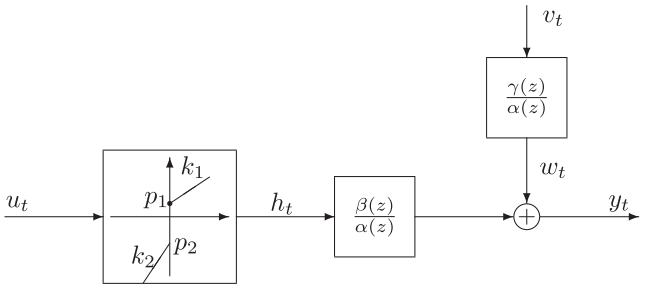

FIGURE 1. The Hammerstein model with asymmetric preload nonlinearity.

II. SYSTEM DESCRIPTION AND IDENTIFICATION MODEL

Let us define some notation first. "$R =: X$" or "$X := R$" stands for "R is defined as X". The superscript T denotes the matrix transpose. $\hat{\boldsymbol{\vartheta}}_t$ stands for the estimate of $\boldsymbol{\vartheta}$ at the sampling time instant t. $\hat{\boldsymbol{\vartheta}}^s$ stands for the estimate of $\boldsymbol{\vartheta}$ at iteration s. z^{-1} stands for an unit backward shift operator: $z^{-1}y_t = y_{t-1}$. $\lambda_{\max}[\boldsymbol{X}]$ denotes the maximum eigenvalue of symmetric square matrix $\boldsymbol{X}$.

Consider a Hammerstein system which consists of one input static nonlinear element and a linear controlled autoregressive moving average (CARMA) subsystems as shown in Fig. 1. The model of the system described in Fig. 1 which disturbed by a moving average noise can be written as

$$\alpha(z)y_t = \beta(z)h(u_t) + \gamma(z)v_t, \tag{1}$$

$$h(u_t) = \begin{cases} k_1 u_t + p_1, & u_t \geqslant 0, \\ k_2 u_t + p_2, & u_t < 0, \end{cases} \tag{2}$$

where u_t is the system input, y_t is the measured output, v_t is white noise with zero mean and variance σ^2, $h(u_t)$ is the output of the nonlinear block, and $\alpha(z)$, $\beta(z)$, and $\gamma(z)$ are polynomials in the unit backward shift operator z^{-1} ($z^{-1}y_t = y_{t-1}$)):

$$\begin{aligned} \alpha(z) &:= 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots + \alpha_{n_\alpha} z^{-n_\alpha}, \\ \beta(z) &:= \beta_1 z^{-1} + \beta_2 z^{-2} + \cdots + \beta_{n_\beta} z^{-n_\beta}, \\ \gamma(z) &:= 1 + \gamma_1 z^{-1} + \gamma_2 z^{-2} + \cdots + \gamma_{n_\gamma} z^{-n_\gamma}. \end{aligned}$$

Then substituting the polynomials $\alpha(z)$, $\beta(z)$, and $\gamma(z)$ into (1) gives

$$\begin{aligned} y_t &= [1 - \alpha(z)]y_t + \beta(z)h(u_t) + \gamma(z)v(t) \\ &= -\alpha_1 y_{t-1} - \alpha_2 y_{t-2} - \cdots - \alpha_{n_\alpha} y_{t-n_\alpha} \\ &\quad + \beta_1 h(u_{t-1}) + \beta_2 h(u_{t-2}) + \cdots + \beta_{n_\beta} h(u_{t-n_\beta}) \\ &\quad + \gamma_1 v_{t-1} + \gamma_2 v_{t-2} + \cdots + \gamma_{n_\gamma} v_{t-n_\gamma} + v_t. \end{aligned} \tag{3}$$

Here we assume that the order n_α, n_β and n_γ are known and consider that $y_t = 0$, $u_t = 0$ and $v_t = 0$ for $t \leqslant 0$. From (3), we can see that there exists the product terms of parameter β_i

and the output of nonlinear block $h(u_t)$. However, the nonlinear output $h(u_t)$ has the asymmetric preload nonlinearity, which is described by the piecewise functions with unknown slopes and preload points. The unknown parameters that need to be identified includes $\{\alpha_i, \beta_i, \gamma_i, k_1, k_2, p_1, p_2\}$. So it is difficult to define one parameter vector to contains all the unknown parameters. In order to overcome this difficulty, we employ a switching function to re-describe the form of the nonlinear block.

Define a switching function

$$g(x) = \begin{cases} 1, & x \geqslant 0, \\ 0, & x < 0. \end{cases}$$

Then the output $h(u_t)$ can be expressed in a new form

$$h(u_t) = (k_1 u_t + p_1)g(u_t) + (k_2 u_t + p_2)g(-u_t). \tag{4}$$

Substituting (4) into (3) gives

$$\begin{aligned} y_t =& -\alpha_1 y_{t-1} - \alpha_2 y_{t-2} - \cdots - \alpha_{n_\alpha} y_{t-n_\alpha} \\ &+ \beta_1 k_1 u_{t-1} g(u_{t-1}) + \beta_1 p_1 g(u_{t-1}) + \beta_1 k_2 u_{t-1} g(-u_{t-1}) \\ &+ \beta_1 p_2 g(-u_{t-1}) + \beta_2 k_1 u_{t-2} g(u_{t-2}) + \beta_2 p_1 g(u_{t-2}) \\ &+ \beta_2 k_2 u_{t-2} g(-u_{t-2}) + \beta_2 p_2 g(-u_{t-2}) + \cdots \\ &+ \beta_{n_\beta} k_1 u_{t-n_\beta} g(u_{t-n_\beta}) + \beta_{n_\beta} p_1 g(u_{t-n_\beta}) \\ &+ \beta_{n_\beta} k_2 u_{t-n_\beta} g(-u_{t-n_\beta}) + \beta_{n_\beta} p_2 g(-u_{t-n_\beta}) \\ &+ \gamma_1 v_{t-1} + \gamma_2 v_{t-2} + \cdots + \gamma_{n_\gamma} v_{t-n_\gamma} + v_t. \end{aligned} \tag{5}$$

It can be observed that there are several product terms of parameters in (5) which make it impossible to obtain the unique parameter estimates. Therefore, at least one parameter in two blocks should be fixed. For tractability of parameter estimation, the key term separation technique was employed to simplify the mathematical form. According to the decomposition approach studied in [28] and [29], we separate a key term $h(u_{t-1})$ and assume $\beta_1 = 1$. Then, the output function can be rewritten as

$$\begin{aligned} y_t =& [1 - \alpha(z)]y_t + h(u_{t-1}) + [\beta(z) - z^{-1}]h(u_t) + \gamma(z)v(t) \\ =& -\alpha_1 y_{t-1} - \alpha_2 y_{t-2} - \cdots - \alpha_{n_\alpha} y_{t-n_\alpha} + k_1 u_{t-1} g(u_{t-1}) \\ &+ p_1 g(u_{t-1}) + k_2 u_{t-1} g(-u_{t-1}) + p_2 g(-u_{t-1}) \\ &+ \beta_2 h(u_{t-2}) + \cdots + \beta_{n_\beta} h(u_{t-n_\beta}) + \gamma_1 v_{t-1} \\ &+ \gamma_2 v_{t-2} + \cdots + \gamma_{n_\gamma} v_{t-n_\gamma} + v_t. \end{aligned} \tag{6}$$

Let $n := n_\alpha + n_\beta + n_\gamma + 3$. Define the parameter vector $\boldsymbol{\vartheta}$ and the information vector $\boldsymbol{\psi}_t$ as

$$\begin{aligned} \boldsymbol{\vartheta} :=& [\alpha_1, \alpha_2, \ldots, \alpha_{n_\alpha}, k_1, p_1, k_2, p_2, \beta_2, \ldots, \beta_{n_\beta}, \\ & \gamma_1, \gamma_2, \ldots, \gamma_{n_\gamma}]^{\mathrm{T}} \in \mathbb{R}^n, \\ \boldsymbol{\psi}_t :=& [-y_{t-1}, -y_{t-2}, \ldots, -y_{t-n_\alpha}, u_{t-1}g(u_{t-1}), g(u_{t-1}), \\ & u_{t-1}g(-u_{t-1}), g(-u_{t-1}), h(u_{t-2}), \ldots, h(u_{t-n_\beta}), \\ & v_{t-1}, v_{t-2}, \ldots, v_{t-n_\gamma}]^{\mathrm{T}} \in \mathbb{R}^n. \end{aligned} \tag{7}$$

Then (7) can be rewritten as

$$y_t = \boldsymbol{\psi}_t^{\mathrm{T}} \boldsymbol{\vartheta} + v_t. \tag{8}$$

The aim of the paper is to obtain the estimates of the unknown model parameters in the input nonlinear controlled autoregressive moving average (IN-CARMA) systems by using the measurement data. Based on the linear regression identification model in (8), the next sections will solve the problem of parameter estimation by using the iterative algorithm under some optimal ways.

Remark 1: By separate a key term, the input nonlinear system can be represented by an output equation, in which all the unknown parameters are separated in one parameter vector and the output equation is linear in parameters, nonlinear in intermediate variables.

III. THE STOCHASTIC GRADIENT ALGORITHM AND GRADIENT ITERATIVE ALGORITHM WITH FINITE MEASUREMENT

In order to show the advantages of the proposed sliding window iterative algorithm, this section first gives the stochastic gradient algorithm then derives the gradient iterative algorithm with finite measurement for comparisons.

A. THE STOCHASTIC GRADIENT ALGORITHM

Consider the input-output data set $\{y_j, u_j, 0 \leqslant j \leqslant t\}$, and define a quadratic cost function

$$J(\boldsymbol{\vartheta}) := \sum_{j=1}^{t} [y(j) - \boldsymbol{\psi}^{\mathrm{T}}(j)\boldsymbol{\vartheta}]^2.$$

Minimizing $J(\boldsymbol{\vartheta})$ based on the negative gradient search, we can obtain the stochastic gradient (SG) algorithm for estimating $\boldsymbol{\vartheta}$ in the IN-CARMA system:

$$\hat{\boldsymbol{\vartheta}}_t = \hat{\boldsymbol{\vartheta}}_{t-1} + \frac{\boldsymbol{\psi}_t}{r_t}[y_t - \boldsymbol{\psi}_t^{\mathrm{T}} \hat{\boldsymbol{\vartheta}}(t-1)], \tag{9}$$

$$r_t = r_{t-1} + \|\boldsymbol{\psi}_t\|^2, \quad r_0 = 1, \tag{10}$$

$$\begin{aligned} \boldsymbol{\psi}_t =& [-y_{t-1}, -y_{t-2}, \ldots, -y_{t-n_\alpha}, u_{t-1}g(u_{t-1}), g(u_{t-1}), \\ & u_{t-1}g(-u_{t-1}), g(-u_{t-1}), h(u_{t-2}), \ldots, h(u_{t-n_\beta}), \\ & v_{t-1}, v_{t-2}, \ldots, v_{t-n_\gamma}]^{\mathrm{T}}. \end{aligned} \tag{11}$$

The difficulty in executing the algorithm (9)–(11) is that the information vector $\boldsymbol{\psi}_t$ contains unknown inner variables $h(u_{t-i})$ and noise terms v_{t-j}. Furthermore, to compute the estimates of $h(u_{t-i})$ and v_{t-j}, the estimates of system parameters are needed. To overcome this difficulty, the approach here is to adopt the interactive identification technique. Replace the unknown $h(u_{t-i})$ and v_{t-j} with their estimates obtained at the previous moment.

According to (4), replacing the unknown k_1, k_2, p_1, and p_2 with their estimates $\hat{k}_{1_t}, \hat{k}_{2_t}, \hat{p}_{1_t}$, and $\hat{p}_{2_t}$ gives the estimate of $h(u_t)$:

$$\hat{h}(u_t) = (\hat{k}_{1_t} u_t + \hat{p}_{1_t})g(u_t) + (\hat{k}_{2_t} u_t + \hat{p}_{2_t})g(-u_t). \tag{12}$$

The estimate of v_t can be computed by replacing $\boldsymbol{\psi}_t$ and $\boldsymbol{\vartheta}$ with $\hat{\boldsymbol{\psi}}_t$ and $\hat{\boldsymbol{\vartheta}}_t$ in (8):

$$\hat{v}_t = y_t - \hat{\boldsymbol{\psi}}_t^{\mathrm{T}} \hat{\boldsymbol{\vartheta}}_t. \tag{13}$$

Then redefine the estimated information vector $\hat{\boldsymbol{\psi}}_t$ as

$$\begin{aligned}\hat{\boldsymbol{\psi}}_t = [&-y_{t-1}, -y_{t-2}, \ldots, -y_{t-n_\alpha}, u_{t-1}g(u_{t-1}), g(u_{t-1}),\\ &u_{t-1}g(-u_{t-1}), g(-u_{t-1}), \hat{h}(u_{t-2}), \ldots, \hat{h}(u_{t-n_\beta}),\\ &\hat{v}_{t-1}, \hat{v}_{t-2}, \ldots, \hat{v}_{t-n_\gamma}]^{\mathrm{T}} \in \mathbb{R}^n. \end{aligned} \tag{14}$$

Replacing the information vector $\boldsymbol{\psi}_t$ in (9)–(10) with its estimate $\hat{\boldsymbol{\psi}}_t$ yields the SG algorithm that can be implemented recursively online:

$$\hat{\boldsymbol{\vartheta}}_t = \hat{\boldsymbol{\vartheta}}_{t-1} + \frac{\hat{\boldsymbol{\psi}}_t}{r_t}[y_t - \hat{\boldsymbol{\psi}}_t^{\mathrm{T}}\hat{\boldsymbol{\vartheta}}(t-1)], \tag{15}$$

$$r_t = r_{t-1} + \|\hat{\boldsymbol{\psi}}_t\|^2, \quad r_0 = 1, \tag{16}$$

$$\begin{aligned}\hat{\boldsymbol{\psi}}_t = [&-y_{t-1}, -y_{t-2}, \ldots, -y_{t-n_\alpha}, u_{t-1}g(u_{t-1}), g(u_{t-1}),\\ &u_{t-1}g(-u_{t-1}), g(-u_{t-1}), \hat{h}(u_{t-2}), \ldots, \hat{h}(u_{t-n_\beta}),\\ &\hat{v}_{t-1}, \hat{v}_{t-2}, \ldots, \hat{v}_{t-n_\gamma}]^{\mathrm{T}},\end{aligned}$$

$$\hat{h}(u_t) = (\hat{s}_{1_t}u_t + \hat{p}_{1_t})g(u_t) + (\hat{s}_{2_t}u_t + \hat{p}_{2_t})g(-u_t), \tag{17}$$

$$\hat{v}_t = y_t - \hat{\boldsymbol{\psi}}_t^{\mathrm{T}}\hat{\boldsymbol{\vartheta}}_t. \tag{18}$$

The procedures of computing $\hat{\boldsymbol{\vartheta}}_t$ with the increasing of t by the SG algorithm are listed as follows.

1) To initialize: Let $t = 1$, and set the initial values: $\hat{\boldsymbol{\vartheta}}_0 = 1_n/p_0$, $\hat{h}(u_{t-i})$ and $\hat{v}_{t-i}$ are random numbers, $r_0 = 1$. p_0 is taken to be a large number, for example $p_0 = 10^6$.
2) Collect the input-output data u_t and y_t, construct $\hat{\boldsymbol{\psi}}_t$ by (17).
3) Compute r_t by (16), update the parameter estimate $\hat{\boldsymbol{\vartheta}}_t$ by (15).
4) Compute $\hat{h}(u_t)$ by (17).
5) Compute $\hat{v}_t$ by (18).
6) Increase t by 1, and go to step 2.

Remark 2: The SG algorithm is a recursive algorithm. The new estimate of parameter vector is equal to the estimate of the parameter vector at the previous moment plus a correction term. A group of newest sampled data and the old date are used to update the correction term. With the sampling time moving forward, the SG algorithm can be implemented online to obtain the new parameter estimates. However, the SG algorithm does not use data efficiently and the convergence speed is slow.

B. THE GRADIENT ITERATIVE ALGORITHM WITH FINITE MEASUREMENT

When a batch of sampled-data are collected from the system, we hope to use this batch of data to update the parameter estimates at the same time. Set the length of the finite measurement as L. Based on the data set $\{y_t, u_t, 1 \leqslant t \leqslant L\}$, define a stacked output vector Y_L and a stacked information matrix Ψ_L as

$$\boldsymbol{Y}_L := \begin{bmatrix} y_L \\ y_{L-1} \\ \vdots \\ y_1 \end{bmatrix} \in \mathbb{R}^L, \quad \boldsymbol{\Psi}_L := \begin{bmatrix} \boldsymbol{\psi}_L^{\mathrm{T}} \\ \boldsymbol{\psi}_{L-1}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\psi}_1^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{L\times n}.$$

Define a quadratic cost function

$$J_2(\boldsymbol{\vartheta}) := \|\boldsymbol{Y}_L - \boldsymbol{\Psi}_L\boldsymbol{\vartheta}\|^2.$$

Let $s = 1, 2, 3, \ldots$ be an iterative variable. Minimizing $J_2(\boldsymbol{\vartheta})$ by using the negative gradient search, we have

$$\begin{aligned}\hat{\boldsymbol{\vartheta}}^s &= \hat{\boldsymbol{\vartheta}}^{s-1} - \frac{\mu^s}{2}\mathrm{grad}[J_2(\hat{\boldsymbol{\vartheta}}^{s-1})]\\ &= \hat{\boldsymbol{\vartheta}}^{s-1} + \mu^s\boldsymbol{\Psi}_L^{\mathrm{T}}[\boldsymbol{Y}_L - \boldsymbol{\Psi}_L\hat{\boldsymbol{\vartheta}}^{s-1}],\end{aligned} \tag{19}$$

where μ^s is the iterative step-size. A conservative choice of μ^s is satisfied [26]

$$0 < \mu^s \leqslant \frac{2}{\lambda_{\max}[\boldsymbol{\Phi}_L^{\mathrm{T}}\boldsymbol{\Phi}_L]}. \tag{20}$$

The difficulty that exists in the algorithm (19)–(20) is that the information matrix $\boldsymbol{\Psi}_L$ contains unknown inner variables $h(u_{t-i})$ and noise terms v_{t-j}. Similarly, replacing the unknown $h(u_{t-i})$ and v_{t-j} with their estimates, redefine the estimated information matrix $\hat{\boldsymbol{\Psi}}_L^s$ as

$$\hat{\boldsymbol{\Psi}}_L^s := \begin{bmatrix} (\hat{\boldsymbol{\psi}}_L^s)^{\mathrm{T}} \\ (\hat{\boldsymbol{\psi}}_{L-1}^s)^{\mathrm{T}} \\ \vdots \\ (\hat{\boldsymbol{\psi}}_t^s)^{\mathrm{T}} \\ \vdots \\ (\hat{\boldsymbol{\psi}}_1^s)^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{L\times n},$$

$$\begin{aligned}\hat{\boldsymbol{\psi}}_t^s = [&-y_{t-1}, -y_{t-2}, \ldots, -y_{t-n_\alpha}, u_{t-1}g(u_{t-1}), g(u_{t-1}),\\ &u_{t-1}g(-u_{t-1}), g(-u_{t-1}), \hat{h}^s(u_{t-2}), \ldots, \hat{h}^s(u_{t-n_\beta}),\\ &\hat{v}_{t-1}^s, \hat{v}_{t-2}^s, \ldots, \hat{v}_{t-n_\gamma}^s]^{\mathrm{T}} \in \mathbb{R}^n.\end{aligned}$$

Then replacing the information matrix $\boldsymbol{\Psi}_L$ in (19)–(20) with its estimate $\hat{\boldsymbol{\Psi}}_L$ gives the gradient iterative (GI) algorithm that can be implemented by applying a batch of data repeatedly offline:

$$\hat{\boldsymbol{\vartheta}}^s = \hat{\boldsymbol{\vartheta}}^{s-1} + \mu^s(\hat{\boldsymbol{\Psi}}_L^s)^{\mathrm{T}}[\boldsymbol{Y}_L - \hat{\boldsymbol{\Psi}}_L^s\hat{\boldsymbol{\vartheta}}^{s-1}], \tag{21}$$

$$0 < \mu^s \leqslant \frac{2}{\lambda_{\max}[(\hat{\Phi}_L^s)^{\mathrm{T}}\hat{\boldsymbol{\Phi}}_L^s]}, \tag{22}$$

$$\hat{\boldsymbol{\Psi}}_L^s = \begin{bmatrix} (\hat{\boldsymbol{\psi}}_L^s)^{\mathrm{T}} \\ (\hat{\boldsymbol{\psi}}_{L-1}^s)^{\mathrm{T}} \\ \vdots \\ (\hat{\boldsymbol{\psi}}_t^s)^{\mathrm{T}} \\ \vdots \\ (\hat{\boldsymbol{\psi}}_1^s)^{\mathrm{T}} \end{bmatrix}, \quad 1 \leqslant t \leqslant L \tag{23}$$

$$\begin{aligned}\hat{\boldsymbol{\psi}}_t^s = [&-y_{t-1}, -y_{t-2}, \ldots, -y_{t-n_\alpha}, u_{t-1}g(u_{t-1}), g(u_{t-1}),\\ &u_{t-1}g(-u_{t-1}), g(-u_{t-1}), \hat{h}^s(u_{t-2}), \ldots,\\ &\hat{h}^s(u_{t-n_\beta}), \hat{v}_{t-1}^s, \hat{v}_{t-2}^s, \ldots, \hat{v}_{t-n_\gamma}^s]^{\mathrm{T}},\end{aligned} \tag{24}$$

$$\hat{h}^s(u_t) = (\hat{k}_1^s u_t + \hat{p}_1^s)g(u_t) + (\hat{k}_2^s u_t + \hat{p}_2^s)g(-u_t), \tag{25}$$

$$\hat{v}_t^s = y_t - \hat{\boldsymbol{\psi}}_t^{\mathrm{T}}\hat{\boldsymbol{\vartheta}}^s. \tag{26}$$

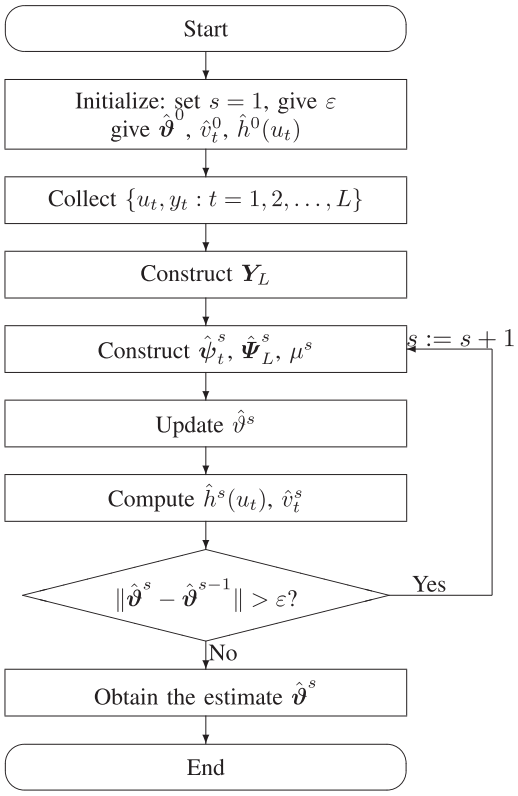

FIGURE 2. The flowchart of the GI algorithm with finite measurement data for computing $\hat{\vartheta}^s$.

The flowchart of computing the parameter estimate $\hat{\vartheta}^s$ in the GI algorithm in (21)–(26) with finite measurement data is shown in Fig. 2.

Remark 3: In the iterative calculation process of each step, the GI algorithm uses all the measured data at the same time, and makes efficient use of the measurement data at each iteration. However, the GI algorithm needs to collect a batch of data first and is implemented offline.

IV. THE SLIDING WINDOW GRADIENT ITERATIVE ALGORITHM

Based on the derived SG algorithm and GI algorithm presented in Section III, a nature question is how can we combine the advantages of the two algorithms to study an approach which can be used for online identification and make sufficient use of the measured data. The idea of sliding window arises. Design a dynamic sliding window to contain a batch of new collected data and use this batch of data to implement the iterative algorithm. Let the length of the dynamic data window as L. When a newest group of system data are collected, the oldest group of data will be discarded. The dynamic data window moves forward with a constant length.

Define a stacked output vector $Y_{t,L}$ and a stacked information matrix $\Psi_{t,L}$ as

$$\boldsymbol{Y}_{t,L} := \begin{bmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-L+1} \end{bmatrix} \in \mathbb{R}^L, \quad \boldsymbol{\Psi}_{t,L} := \begin{bmatrix} \boldsymbol{\psi}_t^{\mathrm{T}} \\ \boldsymbol{\psi}_{t-1}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\psi}_{t-L+1}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{L\times n}.$$

Define a quadratic cost function

$$J_3(\vartheta) := \|\boldsymbol{Y}_{t,L} - \boldsymbol{\Psi}_{t,L}\vartheta\|^2.$$

Let $\hat{\vartheta}_t^s$ be the estimate of $\boldsymbol{\theta}$ at iteration s and time instant t. Minimizing $J_3(\vartheta)$ by using the negative gradient search, we have

$$\begin{aligned}\hat{\vartheta}_t^s &= \hat{\vartheta}_t^{s-1} - \frac{\mu_t^s}{2}\mathrm{grad}[J_3(\hat{\vartheta}_t^{s-1})] \\ &= \hat{\vartheta}_t^{s-1} + \mu_t^s \boldsymbol{\Psi}_{t,L}^{\mathrm{T}}[\boldsymbol{Y}_{t,L} - \boldsymbol{\Psi}_{t,L}\hat{\vartheta}_t^{s-1}],\end{aligned} \tag{27}$$

where μ_t^s is the iterative step-size and satisfies

$$0 < \mu_t^s \leqslant \frac{2}{\lambda_{\max}[\boldsymbol{\Phi}_{t,L}^{\mathrm{T}}\boldsymbol{\Phi}_{t,L}]}. \tag{28}$$

However, similar problems arise. The information matrix $\boldsymbol{\Psi}_{t,L}$ contains the unknown terms $h(u_{t-i})$ and the unmeasured noise terms v_{t-j}. Equation (27) cannot compute the estimate $\hat{\boldsymbol{\theta}}_t^s$ directly. Using the similar approach as in GI algorithm, replace the unknown $h(u_{t-i})$ and v_{t-j} with their estimates. Redefine the estimated information matrix $\hat{\boldsymbol{\Psi}}_{t,L}^s$ as

$$\hat{\boldsymbol{\Psi}}_{t,L}^s := \begin{bmatrix} (\hat{\boldsymbol{\psi}}_t^s)^{\mathrm{T}} \\ (\hat{\boldsymbol{\psi}}_{t-1}^s)^{\mathrm{T}} \\ \vdots \\ (\hat{\boldsymbol{\psi}}_{t-L+1}^s)^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{L\times n},$$

$$\begin{aligned}\hat{\boldsymbol{\psi}}_t^s = [&-y_{t-1}, -y_{t-2}, \ldots, -y_{t-n_\alpha}, u_{t-1}g(u_{t-1}), g(u_{t-1}), \\ &u_{t-1}g(-u_{t-1}), g(-u_{t-1}), \hat{h}^s(u_{t-2}), \ldots, \hat{h}^s(u_{t-n_\beta}), \\ &\hat{v}_{t-1}^s, \hat{v}_{t-2}^s, \ldots, \hat{v}_{t-n_\gamma}^s]^{\mathrm{T}} \in \mathbb{R}^n.\end{aligned}$$

Based on (4), replacing the unknown k_1, k_2, p_1, and p_2 with their estimates $\hat{k}_{1_t}^s$, $\hat{k}_{2_t}^s$, $\hat{p}_{1_t}^s$, and $\hat{p}_{2_t}^s$ gives the estimate of $h(u_t)$:

$$\hat{h}^s(u_t) = (\hat{k}_{1_t}^s u_t + \hat{p}_{1_t}^s)g(u_t) + (\hat{k}_{2_t}^s u_t + \hat{p}_{2_t}^s)g(-u_t). \tag{29}$$

The estimate of v_t^s can be computed by

$$\hat{v}_t^s = y_t - (\hat{\boldsymbol{\psi}}_t^s)^{\mathrm{T}}\hat{\vartheta}_t^s. \tag{30}$$

According to the above derivations, we can summarize the sliding window gradient iterative (SW-GI) identification algorithm for the IN-CARMA systems using a new batch of collect data:

$$\hat{\vartheta}_t^s = \hat{\vartheta}_t^{s-1} + \mu_t^s(\hat{\boldsymbol{\Psi}}_{t,L}^s)^{\mathrm{T}}[\boldsymbol{Y}_{t,L} - \hat{\boldsymbol{\Psi}}_{t,L}^s\hat{\vartheta}_t^{s-1}], \tag{31}$$

$$0 < \mu_t^s \leqslant \frac{2}{\lambda_{\max}[(\hat{\boldsymbol{\Phi}}_{t,L}^s)^{\mathrm{T}}\hat{\boldsymbol{\Phi}}_{t,L}^s]}, \tag{32}$$

$$\hat{\boldsymbol{\Psi}}_{t,L}^s = \begin{bmatrix} (\hat{\boldsymbol{\psi}}_t^s)^{\mathrm{T}} \\ (\hat{\boldsymbol{\psi}}_{t-1}^s)^{\mathrm{T}} \\ \vdots \\ (\hat{\boldsymbol{\psi}}_{t-L+1}^s)^{\mathrm{T}} \end{bmatrix}, \quad t \geqslant L \tag{33}$$

$$\begin{aligned}\hat{\boldsymbol{\psi}}_t^s = [&-y_{t-1}, -y_{t-2}, \ldots, -y_{t-n_\alpha}, u_{t-1}g(u_{t-1}), g(u_{t-1}), \\ &u_{t-1}g(-u_{t-1}), g(-u_{t-1}), \hat{h}^s(u_{t-2}), \ldots, \\ &\hat{h}^s(u_{t-n_\beta}), \hat{v}_{t-1}^s, \hat{v}_{t-2}^s, \ldots, \hat{v}_{t-n_\gamma}^s]^{\mathrm{T}},\end{aligned} \tag{34}$$

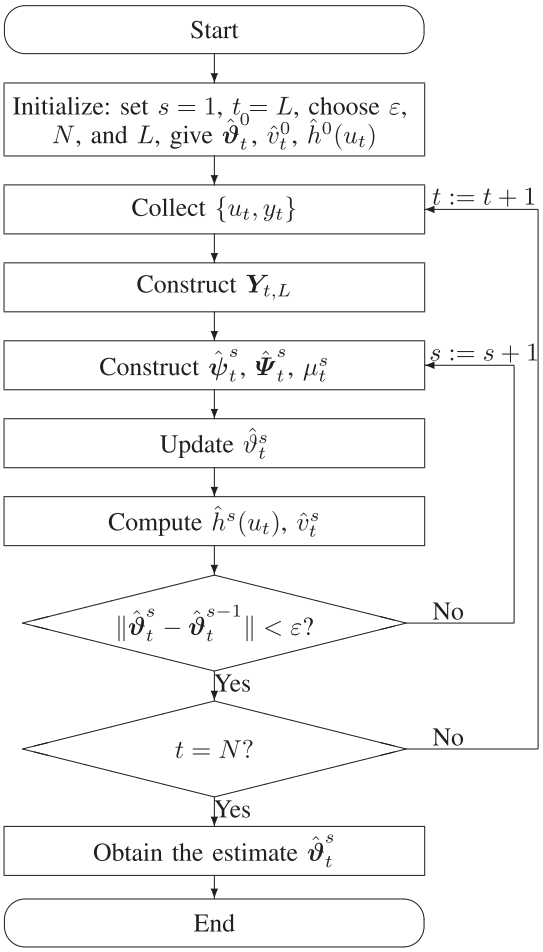

FIGURE 3. The flowchart of the SW-GI algorithm for computing $\hat{\vartheta}_t^s$.

$$\hat{h}^s(u_t) = (\hat{k}_{1,t}^s u_t + \hat{p}_{1,t}^s)g(u_t) + (\hat{k}_{2,t}^s u_t + \hat{p}_{2,t}^s)g(-u_t), \quad (35)$$

$$\hat{v}_t^s = y_t - \hat{\boldsymbol{\psi}}_t^{\mathrm{T}} \hat{\boldsymbol{\vartheta}}_t^s. \quad (36)$$

The flowchart of computing the parameter estimate $\hat{\boldsymbol{\vartheta}}_t^s$ in the SW-GI algorithm in (31)–(36) with a batch of moving-window data is shown in Fig. 3, where N denotes the length of the total database.

Remark 4: In the SW-GI algorithm, a batch of measured data are used to update the parameter estimation at the same time. With the sampling time t increasing, the algorithm can collects new sampled data and can be implemented online.

V. SIMULATION EXAMPLES

Consider a Hammerstein CARMA system with asymmetric preload nonlinearity:

$$\alpha(z)y_t = \beta(z)h(u_t) + \gamma(z)v_t,$$

$$h(u_t) = \begin{cases} k_1 u_t + p_1, & u_t \geqslant 0, \\ k_2 u_t + p_2, & u_t < 0, \end{cases}$$

$$\alpha(z) = 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} = 1 + 0.26z^{-1} - 0.45z^{-2},$$

$$\beta(z) = z^{-1} + \beta_2 z^{-2} = z^{-1} + 0.22z^{-2},$$

$$\gamma(z) = 1 + \gamma_1 z^{-1} = 1 + 0.15z^{-1},$$

$$k_1 = 1, \quad p_1 = 1.5, \; k_2 = 3, \; p_2 = -1.$$

The parameter vector to be estimated is

$$\boldsymbol{\vartheta} = [\alpha_1, \alpha_2, k_1, p_1, k_2, p_2, \beta_2, \gamma_1]^{\mathrm{T}}.$$

The simulation is done by the software of MATLAB. In the example, the input $\{u_t\}$ is taken as a persistent excitation sequence with zero mean and unit variance, $\{v_t\}$ is taken as a Gaussian white noise sequences with zero mean and variance $\sigma^2 = 0.10^2$, $\sigma^2 = 0.50^2$ and $\sigma^2 = 1.00^2$. The output $\{y_t\}$ is generated through simulation of the above specified model.

Applying the SG algorithm, GI algorithm and the SW-GI algorithm to estimate the parameters of this system, respectively. Under the same noise variance $\sigma^2 = 0.50^2$, the parameters and the estimation errors ($\delta := \|\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}\|/\|\boldsymbol{\vartheta}\|$) are shown in Table 1, where the data length of GI and the window length of SW-GI are both set as $L = 1000$. In the SW-GI algorithm, the symbol s represents an iterative variable and k represents a recursive variable, the length of the sliding window is 900.

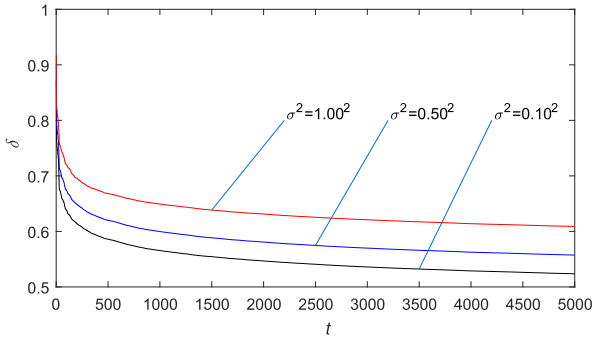

FIGURE 4. The SG estimation errors versus t with different noise variances.

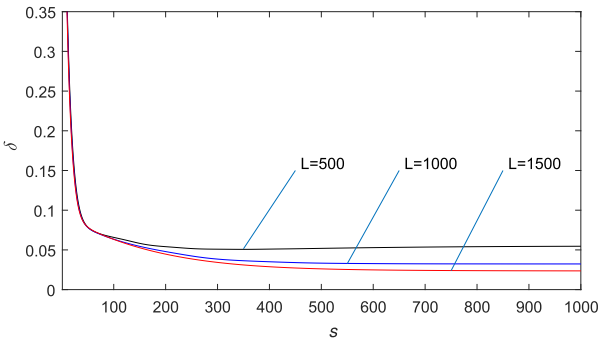

FIGURE 5. The GI estimation errors versus s with different data lengths ($\sigma^2 = 0.50^2$).

Fig. 4 shows the estimation errors δ versus t with different noise variances under the SG algorithm. The GI estimation errors δ versus s with different data lengths are presented in Fig. 5. In SW-GI algorithm, taking the length of the data window as 500 and letting the data window slides forward 50 times, the estimation errors δ versus s and k are shown in Fig. 6. Furthermore, increasing the length of the data window to 1000 and iteration to 100, the estimation errors are presented in Fig. 7. Obviously, Fig. 7 shows a faster converge speed. Under the same noise variance, changing the length of

TABLE 1. The SW-GI parameter estimates and errors with different data lengths and iterations.

Algorithms	$t/s/k$	a_1	a_2	b_2	k_1	p_1	k_2	p_2	d_1	δ (%)
SG	100	0.32252	-0.19135	0.17514	0.59896	0.58104	0.24968	-0.05235	0.23771	67.34463
	500	0.36348	-0.29646	0.17045	0.65332	0.63458	0.31807	-0.11843	0.23501	62.00511
	1000	0.35329	-0.32112	0.17539	0.67383	0.65413	0.34669	-0.14526	0.24747	59.97114
	2000	0.34478	-0.34161	0.17825	0.69405	0.67403	0.37253	-0.17024	0.25628	58.08047
	5000	0.33508	-0.36219	0.18263	0.72183	0.69994	0.40392	-0.20026	0.26624	55.73023
GI	1	-0.26310	-1.37667	0.75129	0.88949	0.87049	1.32768	0.65703	0.98498	77.02194
	50	0.25893	-0.45807	0.19295	1.18917	1.30277	2.99330	-0.94151	0.14199	7.62471
	100	0.26345	-0.44655	0.21017	1.14360	1.33768	3.05097	-0.97704	0.17415	6.11347
	200	0.26412	-0.44595	0.21236	1.08615	1.39095	3.06715	-0.96410	0.19755	4.49537
	500	0.26905	-0.45106	0.20696	1.01886	1.45854	3.08442	-0.94525	0.19566	3.27061
SW-GI	1	0.26501	-0.44368	0.22085	1.10378	1.37896	2.50477	-1.50913	0.22006	19.83780
	2	0.26118	-0.44598	0.21924	1.04206	1.44838	2.71269	-1.29498	0.21216	11.44003
	10	0.25422	-0.44970	0.21592	0.94988	1.54713	3.00605	-0.98987	0.18470	2.12328
	50	0.25370	-0.44927	0.21614	0.95126	1.55059	3.00010	-1.00049	0.18128	2.09550
	100	0.25319	-0.44980	0.21664	0.96047	1.55190	3.00923	-0.99827	0.17736	1.94684
True values		0.26000	-0.45000	0.22000	1.00000	1.50000	3.00000	-1.00000	0.15000	

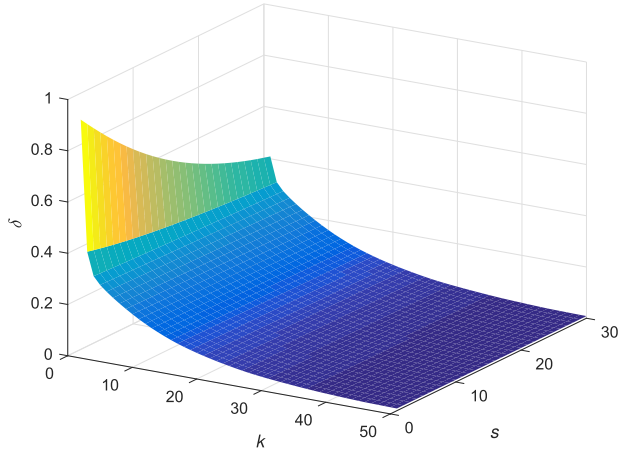

FIGURE 6. The SW-GI estimation errors against k and s ($L = 500$).

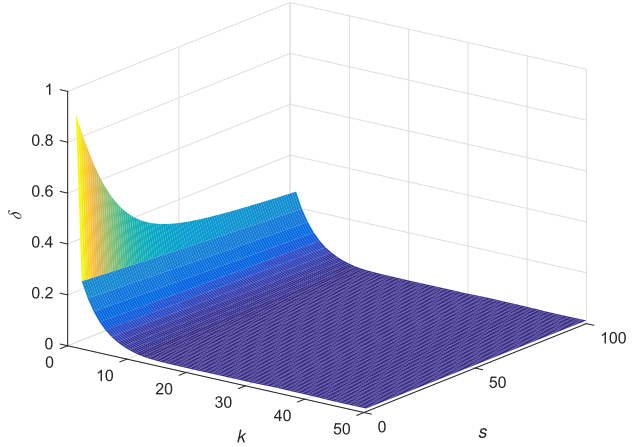

FIGURE 7. The SW-GI estimation errors against k and s ($L = 1000$).

the data window and the number of the iteration, the different parameter estimates are displayed in Table 2.

For model validation, we adopt a different batch of data ($L_t = 1000$ samples from $t = 2001$ to 3000) to compute the root of the mean square errors for the predicted outputs.

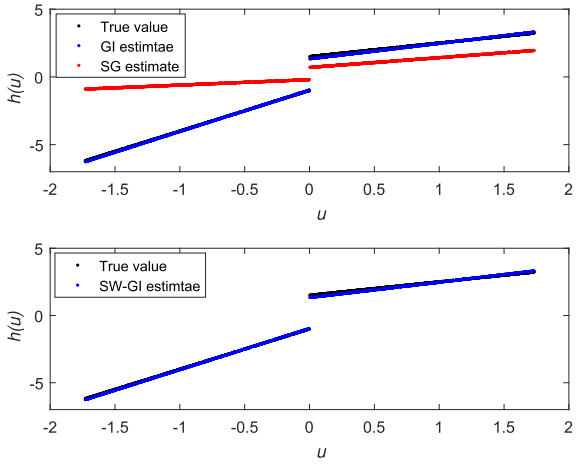

FIGURE 8. The true nonlinearity and estimated proload functions under different algorithms.

The estimated models are constructed by using the parameter estimates shown in Table 1. The true preload function and the estimated preload function $\hat{h}(u_t)$ are plotted in Figure 8. The true outputs and the predicted outputs are presented in Fig. 9. The root-mean-square errors (RMSE) are computed by

$$\delta_1 = \sqrt{\frac{1}{L_t}\sum_{j=2001}^{3000}[y_j - \hat{y}_{\mathrm{sg}_j}]^2} = 2.60780,$$

$$\delta_2 = \sqrt{\frac{1}{L_t}\sum_{j=2001}^{3000}[y_j - \hat{y}_{\mathrm{gi}_j}]^2} = 0.50553,$$

$$\delta_3 = \sqrt{\frac{1}{L_t}\sum_{j=2001}^{3000}[y_j - \hat{y}_{\mathrm{swgi}_j}]^2} = 0.50005,$$

where y_j is the true output; $\hat{y}_{\mathrm{sg}_j}$, $\hat{y}_{\mathrm{gi}_j}$ and $\hat{y}_{\mathrm{swgi}_j}$ are the predicted outputs by using the SG algorithm, GI algorithm, and SW-GI algorithm, respectively.

TABLE 2. The SW-GI parameter estimates and errors with different data lengths and iterations.

L, S	k	a_1	a_2	b_2	k_1	p_1	k_2	p_2	d_1	δ (%)
$L = 500, s = 30$	1	0.23841	-0.46974	0.16203	0.89014	0.96818	1.62413	-1.28898	0.23280	40.98064
	2	0.27921	-0.44894	0.22769	1.09903	1.21711	2.07926	-1.60850	0.23948	31.12076
	5	0.29153	-0.44126	0.24197	1.15702	1.34021	2.40816	-1.59567	0.26276	23.79551
	10	0.29224	-0.44022	0.24438	1.12678	1.37750	2.62173	-1.40223	0.26126	16.04899
	20	0.28814	-0.44258	0.24364	1.08364	1.42041	2.84721	-1.17546	0.23151	7.45380
	50	0.28679	-0.43810	0.24437	1.05062	1.44805	3.03838	-0.98905	0.19388	2.74376
$L = 1000, s = 30$	1	0.23464	-0.46511	0.16662	0.91993	0.98206	1.67653	-1.29814	0.22647	39.53883
	2	0.27590	-0.44527	0.22609	1.11760	1.22073	2.12282	-1.60567	0.21454	30.11493
	5	0.28374	-0.43909	0.23161	1.15682	1.34533	2.42683	-1.59314	0.21333	23.23420
	10	0.28212	-0.44038	0.23196	1.10863	1.39819	2.61898	-1.40745	0.21541	15.77852
	20	0.27905	-0.44180	0.23185	1.05030	1.45824	2.82995	-1.19120	0.20700	7.35724
	50	0.27430	-0.44376	0.22966	1.00488	1.51275	3.00345	-1.00900	0.20104	1.53944
$L = 1000, s = 100$	1	0.26430	-0.44247	0.21833	1.14554	1.30315	2.32019	-1.63810	0.18855	26.17001
	2	0.26413	-0.44239	0.21816	1.11801	1.36504	2.50504	-1.51126	0.20363	19.95997
	5	0.26040	-0.44482	0.21711	1.04036	1.44467	2.77859	-1.23091	0.19376	8.95477
	10	0.25706	-0.44671	0.21533	0.99241	1.49323	2.94029	-1.06294	0.18207	2.53015
	20	0.25462	-0.44799	0.21253	0.97515	1.51646	2.98647	-1.01411	0.17611	1.22548
	50	0.25317	-0.44821	0.21207	0.98779	1.51150	2.97368	-1.02666	0.16743	1.24351
True values		0.26000	-0.45000	0.22000	1.0000	1.50000	3.00000	-1.00000	0.15000	

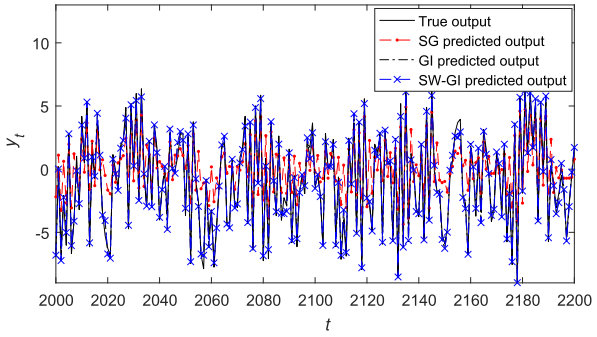

FIGURE 9. The true output and predicted output under different algorithms.

From Tables 1–2 and Figs. 4–9, the following conclusions can be drawn.

- The parameter estimation errors of the SG algorithm are becoming smaller as t increasing and with the noise variance decreasing, the accuracy of the parameter estimation is improved – see the estimation error curves in Fig. 4.
- As the iteration s increasing, the parameter estimation errors of the GI algorithm are decreasing. Compared with the SG algorithm, under the same data length, the GI algorithm can generates more accurate parameter estimates – see the estimation errors in Table 1.
- In the GI algorithm, under the same number of iterations, the longer the data length, the smaller the parameter estimation error obtained – see the estimation error curves in Fig. 5.
- The parameter estimation errors of the SW-GI algorithm are becoming smaller as s and k increasing. The SW-GI algorithm can generates more accurate parameter estimates than the SG and GI algorithm with smaller data length and fewer iterations – see the estimation errors in Table 1.
- In the SW-GI algorithm, increasing the number of iterations while keeping the window size constant, or increasing the length of the data window by a fixed number of iterations can improve the accuracy of the parameter estimation – see the estimation errors in Table 2 and the estimation error curves in Figs. 6–7.
- The curves of the estimated nonlinear functions are very close to the true ones – see Fig. 8. The SW-GI predicted outputs show good fitness with the true outputs, which indicates the effectiveness of the proposed SW-GI algorithm – see Fig. 9.

VI. CONCLUSIONS

This paper mainly presents a sliding window based gradient iterative algorithm for the Hammerstein systems with asymmetric preload nonlinearity based on the key term separation technique. The linear dynamic subsystem is described by a CARMA process. Compared with the SG algorithm, although the SW-GI algorithm has a heavier computational load, it can significantly improve the accuracy of the parameter estimation. Within each set of data window in SW-GI algorithm, the GI algorithm is first used to obtain a reliable parameter estimate, and then as the data window moving, the new measured data is applied to further update the parameter estimates. So the SW-GI algorithm has a faster convergence speed than the SG and GI algorithm. The results of the simulation example demonstrates that the proposed SW-GI algorithm can generates effective parameter estimates and reliable model predictions.

REFERENCES

[1] F. Ding, F. Wang, L. Xu, T. Hayat, and A. Alsaedi, "Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique," *IET Control Theory Appl.*, vol. 11, no. 3, pp. 390–400, Feb. 2017.

[2] F. Ding, L. Xu, and Q. Zhu, "Performance analysis of the generalised projection identification for time-varying systems," *IET Control Theory Appl.*, vol. 10, no. 18, pp. 2506–2514, Dec. 2016.
[3] M. Li and X. Liu, "Auxiliary model based least squares iterative algorithms for parameter estimation of bilinear systems using interval-varying measurements," *IEEE Access*, vol. 6, pp. 21518–21529, 2018.
[4] J. Chen, Q. M. Zhu, J. Li, and Y. J. Liu, "Biased compensation recursive least squares-based threshold algorithm for time-delay rational models via redundant rule," *Nonlinear Dyn.*, vol. 91, no. 2, pp. 797–807, Jan. 2018.
[5] J. Pan, H. Ma, X. Jiang, W. Ding, and F. Ding, "Adaptive gradient-based iterative algorithm for multivariable controlled autoregressive moving average systems using the data filtering technique," *Complexity*, vol. 2018, Jul. 2018, Art. no. 9598307. doi: 10.1155/2018/9598307.
[6] J. Pan, W. Li, and H. Zhang, "Control algorithms of magnetic suspension systems based on the improved double exponential reaching law of sliding mode control," *Int. J. Control Autom. Syst.*, vol. 16, no. 6, pp. 2878–2887, Dec. 2018.
[7] S. Han, J. Kim, and K. Sung, "Extended generalized total least squares method for the identification of bilinear systems," *IEEE Trans. Signal Process.*, vol. 44, no. 4, pp. 1015–1018, Apr. 1996.
[8] G. Vandersteen, "On the use of compensated total least squares in system identification," *IEEE Trans. Autom. Control*, vol. 43, no. 10, pp. 1436–1441, Oct. 1998.
[9] F. Chen, F. Ding, A. Alsaedi, and T. Hayat, "Data filtering based multi-innovation extended gradient method for controlled autoregressive autoregressive moving average systems using the maximum likelihood principle," *Math. Comput. Simul.*, vol. 132, pp. 53–67, Feb. 2017.
[10] L. Xu and F. Ding, "Parameter estimation for control systems based on impulse responses," *Int. J. Control, Autom., Syst.*, vol. 15, no. 6, pp. 2471–2479, Dec. 2017.
[11] L. Xu, "The parameter estimation algorithms based on the dynamical response measurement data," *Adv. Mech. Eng.*, vol. 9, no. 11, pp. 1–12, Nov. 2017.
[12] L. Xu and F. Ding, "Iterative parameter estimation for signal models based on measured data," *Circuits Syst. Signal Process.*, vol. 37, no. 7, pp. 3046–3069, Jul. 2018.
[13] L. Xu, W. Xiong, A. Alsaedi, and T. Hayat, "Hierarchical parameter estimation for the frequency response based on the dynamical window data," *Int. J. Control Autom. Syst.*, vol. 16, no. 4, pp. 1756–1764, Aug. 2018.
[14] Y. Wang and F. Ding, "A filtering based multi-innovation gradient estimation algorithm and performance analysis for nonlinear dynamical systems," *IMA J. Appl. Math.*, vol. 82, no. 6, pp. 1171–1191, Dec. 2017.
[15] Y. Mao and F. Ding, "A novel parameter separation based identification algorithm for Hammerstein systems," *Appl. Math. Lett.*, vol. 60, pp. 21–27, Oct. 2016.
[16] Y. Wang and F. Ding, "Iterative estimation for a non-linear IIR filter with moving average noise by means of the data filtering technique," *IMA J. Math. Control Inf.*, vol. 34, no. 3, pp. 745–764, Sep. 2017.
[17] Y. Bao, L. Y. Wang, C. Wang, and Y. Wang, "Hammerstein models and real-time system identification of load dynamics for voltage management," *IEEE Access*, vol. 6, pp. 34598–34607, Jun. 2018.
[18] F. Ding, Y. Wang, J. Dai, Q. Li, and Q. Chen, "A recursive least squares parameter estimation algorithm for output nonlinear autoregressive systems using the input–output data filtering," *J. Franklin Inst.*, vol. 354, no. 15, pp. 6938–6955, Oct. 2017.
[19] B. Ding and B. Huang, "Output feedback model predictive control for nonlinear systems represented by Hammerstein–Wiener model," *IET Control Theory Appl.*, vol. 1, no. 5, pp. 1302–1310, Sep. 2007.
[20] H. B. Chen, Y. Xiao, and F. Ding, "Hierarchical gradient parameter estimation algorithm for Hammerstein nonlinear systems using the key term separation principle," *Appl. Math. Comput.*, vol. 247, pp. 1202–1210, Nov. 2014.
[21] E.-W. Bai, "An optimal two-stage identification algorithm for Hammerstein–Wiener nonlinear systems," *Automatica*, vol. 34, pp. 333–338, Mar. 1998.
[22] X. Zhang, F. Ding, A. Alsaadi, and T. Hayat, "Recursive parameter identification of the dynamical models for bilinear state space systems," *Nonlinear Dyn.*, vol. 89, no. 4, pp. 2415–2429, 2017.
[23] X. Zhang, F. Ding, L. Xu, and E. Yang, "State filtering-based least squares parameter estimation for bilinear systems using the hierarchical identification principle," *IET Control Theory Appl.*, vol. 12, no. 12, pp. 1704–1713, Aug. 2018.
[24] E.-W. Bai, "Identification of linear systems with hard input nonlinearities of known structure," *Automatica*, vol. 38, no. 5, pp. 853–860, May 2002.
[25] J. Vörös, "Iterative algorithm for parameter identification of Hammerstein systems with two-segment nonlinearities," *IEEE Trans. Autom. Control*, vol. 44, no. 11, pp. 2145–2149, Nov. 1999.
[26] F. Ding, L. Xu, F. E. Alsaadi, and T. Hayat, "Iterative parameter identification for pseudo-linear systems with ARMA noise using the filtering technique," *IET Control Theory Appl.*, vol. 12, no. 7, pp. 892–899, May 2018.
[27] S. Zhao, Y. S. Shmaliy, and F. Liu, "On the iterative computation of error matrix in unbiased FIR filtering," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 555–558, May 2017.
[28] J. Vörös, "Recursive identification of Hammerstein systems with discontinuous nonlinearities containing dead-zones," *IEEE Trans. Autom. Control*, vol. 48, no. 12, pp. 2203–2206, Dec. 2003.
[29] J. Vörös, "Modeling and parameter identification of systems with multisegment piecewise-linear characteristics," *IEEE Trans. Autom. Control*, vol. 47, no. 1, pp. 184–188, Jan. 2002.

JUNXIA MA received the M.Sc. and Ph.D. degrees from the School of Internet of Things Engineering, Jiangnan University, Wuxi, China, in 2014 and 2017, respectively, where she is currently a Lecturer. Her current research interests include system identification, parameter estimation, and processing control.

QIULIN FEI received the B.Sc. degree from the School of Internet of Things Engineering, Jiangnan University, in 2018, Wuxi, China, where she is currently pursuing the M.Sc. degree. Her research interests include system identification and state estimation.

WEILI XIONG received the Ph.D. degree from the School of Communication and Control Engineering, Jiangnan University, in 2007, Wuxi, China. She was a Visiting Scholar with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Canada, from 2013 to 2014. She is currently a Professor with the School of Internet of Things Engineering, Jiangnan University. Her research interests include system identification, soft sensor of industry processes, and optimization.