Least squares based and gradient based iterative identification for Wiener nonlinear systems ☆

Dongqing Wang [a,*], Feng Ding [b]

[a] College of Automation Engineering, Qingdao University, Qingdao 266071, PR China
[b] School of IoT Engineering, Jiangnan University, Wuxi 214122, PR China

ARTICLE INFO

ABSTRACT

This paper derives a least squares-based and a gradient-based iterative identification algorithms for Wiener nonlinear systems. These methods separate one bilinear cost function into two linear cost functions, estimating directly the parameters of Wiener systems without re-parameterization to generate redundant estimates. The simulation results confirm that the proposed two algorithms are valid and the least squares-based iterative algorithm has faster convergence rates than the gradient-based iterative algorithm.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Nonlinear block oriented model such as Wiener and Hammerstein models can be used to approximate many nonlinear dynamic process. A Wiener model has a linear dynamic block followed by a static nonlinear function and a Hammerstein model puts a nonlinear part before a linear dynamic one [1–5].

The identification issues for Wiener systems have attracted great attention. Most existing contributions assumed that the nonlinear part of Wiener models is a linear combination or a piecewise-linear function [6,7], or has an inverse function over the operating range of interest [8]. Hu and Chen [9] studied a no-invertibility nonlinear part of Wiener models; Kozek and Sinanović [10] used optimal local linear models for Wiener models identification; Figueroa et al. [11] proposed a simultaneous approach for Wiener model identification. Hagenblad et al. [12] derived a maximum likelihood method to identify Wiener models.

In the field of system modeling and control, the iterative identification methods are usually used to estimate the parameters of linear and nonlinear systems in which the information vector contains unknown variables (unmeasured variables or unknown noise terms) [6,7,13–15]. Kapetanios [16] gave a simple iterative idea for ARMA and VARMA models. Vörös [6,7] used the iterative approaches to identify the parameters of Wiener models. The iterative solution of a bilinear equation system was proposed by Bai [17], this is a method using the hierarchical identification principle [18–20]. Ding and Chen [1] developed an iterative and a recursive least squares algorithms for Hammerstein nonlinear ARMAX systems. Their approaches require estimating more parameters than the Hammerstein system since by re-parameterization, the number of the

parameters to be identified increases, leading to many redundant estimates.

On the basis of the work in [21], this paper derives a least squares-based and a gradient-based iterative algorithms by introducing two cost functions and by using the hierarchical identification principle in [18]. These iterative methods avoid re-parameterizing the linear and nonlinear parts of the Wiener system to generate redundant estimates. They estimate directly the parameters, increasing computation efficiency. These iterative algorithms use all the measured input–output data at each iterative computation (at each iteration), and thus can produce highly accurate parameter estimation. Ding et al. [22–29] presented several novel multi-innovation identification methods which can be applied to the Wiener nonlinear systems in this paper.

Briefly, the paper is organized as follows. Section 2 describes identification problem formulation for the Wiener nonlinear systems. Sections 3 and 4 derive a least squares-based and a gradient-based iterative algorithms for the Wiener systems, respectively. Section 5 provides an illustrative example to show the effectiveness of the proposed algorithms. Finally, we offer some concluding remarks in Section 6.

2. Problem description

Let us introduce some notations first. The symbol $\boldsymbol{I}_n$ stands for an identity matrix of order n and $\mathbf{I}$ is an identity matrix of appropriate sizes; the superscript T denotes the matrix transpose; $\mathbf{1}_n$ represents an n-dimensional column vector whose elements are 1; the norm of a matrix $\boldsymbol{X}$ is defined by $\|\boldsymbol{X}\|^2 = \mathrm{tr}[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}]$; $\lambda_{\max}[\boldsymbol{X}]$ represents the maximum eigenvalue of the square matrix $\boldsymbol{X}$.

Refer to the Wiener models in [8,11] and Hammerstein–Wiener models in [30,31], and consider the following Wiener system with colored noise:

$$y(t) = \sum_{i=1}^{p} a_i \sum_{l=1}^{q} d_l g_l[y(t-i)] + \sum_{j=1}^{n} b_j u(t-j) + \sum_{k=1}^{m} f_k v(t-k) + v(t),$$

(1)

where $u(t)$ and $y(t)$ are the system input and output, respectively, $v(t)$ is a white noise with zero mean, $g_i(*)$ are known nonlinear base functions [30]. Assume that $u(t)=0, y(t)=0$ and $v(t)=0$ for $t \leq 0$ and the orders p, q, n and m are known.

Define the parameter vectors,

$$\boldsymbol{a} := [a_1, a_2, \ldots, a_p]^{\mathrm{T}} \in \mathbb{R}^p,$$

$$\boldsymbol{d} := [d_1, d_2, \ldots, d_q]^{\mathrm{T}} \in \mathbb{R}^q,$$

$$\boldsymbol{b} := [b_1, b_2, \ldots, b_n]^{\mathrm{T}} \in \mathbb{R}^n,$$

$$\boldsymbol{f} := [f_1, f_2, \ldots, f_m]^{\mathrm{T}} \in \mathbb{R}^m,$$

and the input information vector $\boldsymbol{\varphi}(t)$, the noise information vector $\boldsymbol{\psi}(t)$ and the output information matrix $\boldsymbol{G}(t)$ as

$$\boldsymbol{\varphi}(t) := [u(t-1), u(t-2), \ldots, u(t-n)]^{\mathrm{T}} \in \mathbb{R}^n,$$

(2)

$$\boldsymbol{\psi}(t) := [v(t-1), v(t-2), \ldots, v(t-m)]^{\mathrm{T}} \in \mathbb{R}^m,$$

(3)

$$\boldsymbol{G}(t) := \begin{bmatrix} g_1(y(t-1)) & g_2(y(t-1)) & \cdots & g_q(y(t-1)) \\ g_1(y(t-2)) & g_2(y(t-2)) & \cdots & g_q(y(t-2)) \\ \vdots & \vdots & & \vdots \\ g_1(y(t-p)) & g_2(y(t-p)) & \cdots & g_q(y(t-p)) \end{bmatrix} \in \mathbb{R}^{p \times q}.$$

(4)

Eq. (1) can be rewritten as

$$y(t) = \boldsymbol{a}^{\mathrm{T}} \boldsymbol{G}(t) \boldsymbol{d} + \boldsymbol{\varphi}^{\mathrm{T}}(t) \boldsymbol{b} + \boldsymbol{\psi}^{\mathrm{T}}(t) \boldsymbol{f} + v(t).$$

(5)

Note that for model (5), any pair $\alpha \boldsymbol{a}$ and $\boldsymbol{d}/\alpha$ for any nonzero constant α provides the same input–output data. To have identifiability, we adopt the normalization constraint on $\boldsymbol{d}$ for model (5). There are several ways to make the normalization [30,32,33]. Without loss of generality, we adopt the following:

Assumption 1 (*Liu and Bai [32]*). $\|\boldsymbol{d}\| = 1$, and the first nonzero entry of $\boldsymbol{d}$ is positive, i.e., $d_1 > 0$.

Let k=1, 2, 3, … be an iterative variable and

$$\hat{\theta}_k := \begin{bmatrix} \hat{\boldsymbol{a}}_k \\ \hat{\boldsymbol{b}}_k \\ \hat{\boldsymbol{f}}_k \end{bmatrix},$$

and $\hat{\boldsymbol{\vartheta}}_k := \hat{\boldsymbol{d}}_k$ denote the estimates of

$$\theta := \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{f} \end{bmatrix},$$

and $\boldsymbol{\vartheta} := \boldsymbol{d}$, respectively.

Suppose the data length $L \gg p+q+n+m$. Define the cost function,

$$J(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{f}, \boldsymbol{d}) = \sum_{t=1}^{L} [y(t) - \boldsymbol{a}^{\mathrm{T}} \boldsymbol{G}(t) \boldsymbol{d} - \boldsymbol{\varphi}^{\mathrm{T}}(t) \boldsymbol{b} - \boldsymbol{\psi}^{\mathrm{T}}(t) \boldsymbol{f}]^2.$$

(6)

The model in (5) contains the product of two parameter vectors $\boldsymbol{a}$ and $\boldsymbol{d}$, and this cost function J is a bilinear cost function, which makes the identification problem more difficult than that of linear systems. In order to solve this difficulty, we adopt the hierarchical identification principle [18–20] and decompose this bilinear cost function into two linear cost functions $J(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{f}, \hat{\boldsymbol{d}}_{k-1})$ for fixed $\boldsymbol{d} = \hat{\boldsymbol{d}}_{k-1}$ and $J(\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k, \hat{\boldsymbol{f}}_k, \boldsymbol{d})$ for fixed $\boldsymbol{a} = \hat{\boldsymbol{a}}_k$, $\boldsymbol{b} = \hat{\boldsymbol{b}}_k$, $\boldsymbol{f} = \hat{\boldsymbol{f}}_k$. Thus, minimizing the quadratic functions $J(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{f}, \hat{\boldsymbol{d}}_{k-1})$ in θ and $J(\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k, \hat{\boldsymbol{f}}_k, \boldsymbol{d})$ in $\boldsymbol{d}$ is relatively easy. This idea is equivalent to minimizing the following two optimization problems.

- The optimization of $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{f}$:

$$\{\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k, \hat{\boldsymbol{f}}_k\} = \arg\min_{\theta} \sum_{t=1}^{L} [y(t) - \boldsymbol{a}^{\mathrm{T}} \boldsymbol{G}(t) \hat{\boldsymbol{d}}_{k-1} - \boldsymbol{\varphi}^{\mathrm{T}}(t) \boldsymbol{b} - \boldsymbol{\psi}^{\mathrm{T}}(t) \boldsymbol{f}]^2$$

(7)

for fixed $\hat{\boldsymbol{d}}_{k-1}$, and
- The optimization of $\boldsymbol{d}$:

$$\hat{\boldsymbol{d}}_k = \arg\min_{\boldsymbol{d}} \sum_{t=1}^{L} [y(t) - \hat{\boldsymbol{a}}_k^{\mathrm{T}} \boldsymbol{G}(t) \boldsymbol{d} - \boldsymbol{\varphi}^{\mathrm{T}}(t) \hat{\boldsymbol{b}}_k - \boldsymbol{\psi}^{\mathrm{T}}(t) \hat{\boldsymbol{f}}_k]^2$$

(8)

for fixed $\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k$ and $\hat{\boldsymbol{f}}_k$.

This paper uses the least squares and negative gradient search method to solve the optimization problem in (7) to generate the estimates $\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k$ and $\hat{\boldsymbol{f}}_k$, and the optimization problem in (8) to generate the estimate $\hat{\boldsymbol{d}}_k$.

3. The least squares-based iterative algorithm

Introducing two cost functions:

$$J_1(\boldsymbol{\theta}) = J(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{f}, \hat{\boldsymbol{d}}_{k-1})$$
$$= \sum_{t=1}^{L} [y(t) - \boldsymbol{a}^{\mathrm{T}} \boldsymbol{G}(t) \hat{\boldsymbol{d}}_{k-1} - \boldsymbol{\varphi}^{\mathrm{T}}(t) \boldsymbol{b} - \boldsymbol{\psi}^{\mathrm{T}}(t) \boldsymbol{f}]^2,$$

$$J_2(\boldsymbol{\vartheta}) = J(\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k \hat{\boldsymbol{f}}_k, \boldsymbol{d})$$
$$= \sum_{t=1}^{L} [y(t) - \hat{\boldsymbol{a}}_k^{\mathrm{T}} \boldsymbol{G}(t) \boldsymbol{d} - \boldsymbol{\varphi}^{\mathrm{T}}(t) \hat{\boldsymbol{b}}_k - \boldsymbol{\psi}^{\mathrm{T}}(t) \hat{\boldsymbol{f}}_k]^2.$$

Define the stacked output vector $\boldsymbol{Y}(L)$, input information matrix $\boldsymbol{\Phi}(L)$, noise information matrix $\boldsymbol{\Psi}(L)$, and information matrices $\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L)$ and $\boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)$ as

$$\boldsymbol{Y}(L) := \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(L) \end{bmatrix} \in \mathbb{R}^L, \quad \boldsymbol{\Phi}(L) := \begin{bmatrix} \boldsymbol{\varphi}^{\mathrm{T}}(1) \\ \boldsymbol{\varphi}^{\mathrm{T}}(2) \\ \vdots \\ \boldsymbol{\varphi}^{\mathrm{T}}(L) \end{bmatrix} \in \mathbb{R}^{L \times n},$$

$$\boldsymbol{\Psi}(L) := \begin{bmatrix} \boldsymbol{\psi}^{\mathrm{T}}(1) \\ \boldsymbol{\psi}^{\mathrm{T}}(2) \\ \vdots \\ \boldsymbol{\psi}^{\mathrm{T}}(L) \end{bmatrix} \in \mathbb{R}^{L \times m}, \tag{9}$$

$$\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L) := \begin{bmatrix} \hat{\boldsymbol{d}}_{k-1}^{\mathrm{T}} G^{\mathrm{T}}(1) \\ \hat{\boldsymbol{d}}_{k-1}^{\mathrm{T}} G^{\mathrm{T}}(2) \\ \vdots \\ \hat{\boldsymbol{d}}_{k-1}^{\mathrm{T}} G^{\mathrm{T}}(L) \end{bmatrix} \in \mathbb{R}^{L \times p},$$

$$\boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L) := \begin{bmatrix} \hat{\boldsymbol{a}}_k^{\mathrm{T}} G(1) \\ \hat{\boldsymbol{a}}_k^{\mathrm{T}} G(2) \\ \vdots \\ \hat{\boldsymbol{a}}_k^{\mathrm{T}} G(L) \end{bmatrix} \in \mathbb{R}^{L \times q}. \tag{10}$$

Hence, J_1 and J_2 can be rewritten as

$$J_1(\boldsymbol{\theta}) = \| \boldsymbol{Y}(L) - \boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L) \boldsymbol{a} - \boldsymbol{\Phi}(L) \boldsymbol{b} - \boldsymbol{\Psi}(L) \boldsymbol{f} \|^2$$
$$= \| \boldsymbol{Y}(L) - [\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \boldsymbol{\Psi}(L)] \boldsymbol{\theta} \|^2, \tag{11}$$

$$J_2(\boldsymbol{\vartheta}) = \| \boldsymbol{Y}(L) - \boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L) \boldsymbol{d} - \boldsymbol{\Phi}(L) \hat{\boldsymbol{b}}_k - \boldsymbol{\Psi}(L) \hat{\boldsymbol{f}}_k \|^2. \tag{12}$$

To minimize $J_1(\boldsymbol{\theta})$ and $J_2(\boldsymbol{\vartheta})$, let their partial differentials with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ be zero:

$$\mathrm{grad}[J_1(\boldsymbol{\theta})] = \frac{\partial J_1(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2[\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \boldsymbol{\Psi}(L)]^{\mathrm{T}}$$
$$\times \{ \boldsymbol{Y}(L) - [\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \boldsymbol{\Psi}(L)] \boldsymbol{\theta} \} = \boldsymbol{0},$$

$$\mathrm{grad}[J_1(\boldsymbol{\vartheta})] = \frac{\partial J_2(\boldsymbol{\vartheta})}{\partial \boldsymbol{\vartheta}} = -2 \boldsymbol{\Omega}^{\mathrm{T}}(\hat{\boldsymbol{a}}_k, L) [\boldsymbol{Y}(L) - \boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L) \boldsymbol{\vartheta}$$
$$- \boldsymbol{\Phi}(L) \hat{\boldsymbol{b}}_k - \boldsymbol{\Psi}(L) \hat{\boldsymbol{f}}_k] = \boldsymbol{0}.$$

Provided that the input signal is persistently exciting and the related matrices are non-singular, we can obtain the least squares estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$:

$$\hat{\boldsymbol{\theta}}_k = [\boldsymbol{\Xi}^{\mathrm{T}}(\hat{\boldsymbol{d}}_{k-1}, L) \boldsymbol{\Xi}(\hat{\boldsymbol{d}}_{k-1}, L)]^{-1} \boldsymbol{\Xi}^{\mathrm{T}}(\hat{\boldsymbol{d}}_{k-1}, L) \boldsymbol{Y}(L), \tag{13}$$

$$\hat{\boldsymbol{\vartheta}}_k = [\boldsymbol{\Omega}^{\mathrm{T}}(\hat{\boldsymbol{a}}_k, L) \boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)]^{-1} \boldsymbol{\Omega}^{\mathrm{T}}(\hat{\boldsymbol{a}}_k, L) [\boldsymbol{Y}(L) - \boldsymbol{\Phi}(L) \hat{\boldsymbol{b}}_k - \boldsymbol{\Psi}(L) \hat{\boldsymbol{f}}_k], \tag{14}$$

$$\boldsymbol{\Xi}(\hat{\boldsymbol{d}}_{k-1}, L) := [\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \boldsymbol{\Psi}(L)]. \tag{15}$$

However, $\boldsymbol{\psi}(t), t = 1, 2, \ldots, L$, in $\boldsymbol{\Psi}(L)$ contain the unmeasured noise terms $v(t-i)$ so it is impossible to compute the estimates $\hat{\boldsymbol{\theta}}_k$ and $\hat{\boldsymbol{\vartheta}}_k$. The approach here is based on the iterative identification technique or the hierarchical identification principle [18–20]. Let $\hat{v}_k(t-i)$ be the estimate of $v(t-i)$ at iteration k, and $\hat{\boldsymbol{\psi}}_k(t)$ denote the noise information vector $\boldsymbol{\psi}(t)$ obtained by replacing $v(t-i)$ in (3) with $\hat{v}_{k-1}(t-i)$, i.e.,

$$\hat{\boldsymbol{\psi}}_k(t) := [\hat{v}_{k-1}(t-1), \hat{v}_{k-1}(t-2), \ldots, \hat{v}_{k-1}(t-m)]^{\mathrm{T}}. \tag{16}$$

Replacing $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{f}, \boldsymbol{d}$ and $\psi(t)$ in (5) with $\hat{\boldsymbol{a}}_{k-1}, \hat{\boldsymbol{b}}_{k-1} \hat{\boldsymbol{f}}_{k-1}, \hat{\boldsymbol{d}}_{k-1}$ and $\hat{\boldsymbol{\psi}}_k(t)$, respectively, we can compute the estimate $\hat{v}_k(t)$ by

$$\hat{v}_k(t) = y(t) - \hat{\boldsymbol{a}}_{k-1}^{\mathrm{T}} G(t) \hat{\boldsymbol{d}}_{k-1} - \boldsymbol{\varphi}^{\mathrm{T}}(t) \hat{\boldsymbol{b}}_{k-1} + \hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t) \hat{\boldsymbol{f}}_{k-1}. \tag{17}$$

Define

$$\hat{\boldsymbol{\Psi}}_k(L) := \begin{bmatrix} \hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(1) \\ \hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(2) \\ \vdots \\ \hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(L) \end{bmatrix} \in \mathbb{R}^{L \times m}. \tag{18}$$

Replacing $\boldsymbol{\Psi}(L)$ in (14) and (15) with $\hat{\boldsymbol{\Psi}}_k(L)$, we can summary the least squares-based iterative identification algorithm for the Wiener systems (the W-LSI algorithm for short) as follows:

$$\hat{\boldsymbol{\theta}}_k = [\hat{\boldsymbol{\Xi}}^{\mathrm{T}}(\hat{\boldsymbol{d}}_{k-1}, L) \hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)]^{-1} \hat{\boldsymbol{\Xi}}^{\mathrm{T}}(\hat{\boldsymbol{d}}_{k-1}, L) \boldsymbol{Y}(L), \quad k = 1, 2, 3, \ldots, \tag{19}$$

$$\hat{\boldsymbol{\vartheta}}_k = [\boldsymbol{\Omega}^{\mathrm{T}}(\hat{\boldsymbol{a}}_k, L) \boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)]^{-1} \boldsymbol{\Omega}^{\mathrm{T}}(\hat{\boldsymbol{a}}_k, L) [\boldsymbol{Y}(L) - \boldsymbol{\Phi}(L) \hat{\boldsymbol{b}}_k - \hat{\boldsymbol{\Psi}}_k(L) \hat{\boldsymbol{f}}_k], \tag{20}$$

$$\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L) = [\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \hat{\boldsymbol{\Psi}}_k(L)], \tag{21}$$

$$\boldsymbol{Y}(L) = [y(1), y(2), \ldots, y(L)]^{\mathrm{T}}, \tag{22}$$

$$\boldsymbol{\Phi}(L) = [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \ldots, \boldsymbol{\varphi}(L)]^{\mathrm{T}}, \tag{23}$$

$$\hat{\boldsymbol{\Psi}}_k(L) = [\hat{\boldsymbol{\psi}}_k(1), \hat{\boldsymbol{\psi}}_k(2), \ldots, \hat{\boldsymbol{\psi}}_k(L)]^{\mathrm{T}}, \tag{24}$$

$$\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L) = [G(1) \hat{\boldsymbol{d}}_{k-1}, G(2) \hat{\boldsymbol{d}}_{k-1}, \ldots, G(L) \hat{\boldsymbol{d}}_{k-1}]^{\mathrm{T}}, \tag{25}$$

$$\boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L) = [G^{\mathrm{T}}(1) \hat{\boldsymbol{a}}_k, G^{\mathrm{T}}(2) \hat{\boldsymbol{a}}_k, \ldots, G^{\mathrm{T}}(L) \hat{\boldsymbol{a}}_k]^{\mathrm{T}}, \tag{26}$$

$$\boldsymbol{\varphi}(t) = [u(t-1), u(t-2), \ldots, u(t-n)]^{\mathrm{T}}, \quad t = 1, 2, \ldots, L, \tag{27}$$

$$\hat{\boldsymbol{\psi}}_k(t) = [\hat{v}_{k-1}(t-1), \hat{v}_{k-1}(t-2), \ldots, \hat{v}_{k-1}(t-m)]^{\mathrm{T}}, \tag{28}$$

$$\hat{v}_k(t) = y(t) - \hat{\boldsymbol{a}}_{k-1}^{\mathrm{T}} G(t) \hat{\boldsymbol{d}}_{k-1} - \boldsymbol{\varphi}^{\mathrm{T}}(t) \hat{\boldsymbol{b}}_{k-1} + \hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t) \hat{\boldsymbol{f}}_{k-1}, \tag{29}$$

$$G(t) = \begin{bmatrix} g_1(y(t-1)) & g_2(y(t-1)) & \cdots & g_q(y(t-1)) \\ g_1(y(t-2)) & g_2(y(t-2)) & \cdots & g_q(y(t-2)) \\ \vdots & \vdots & & \vdots \\ g_1(y(t-p)) & g_2(y(t-p)) & \cdots & g_q(y(t-p)) \end{bmatrix}. \quad (30)$$

The computation procedure of the W-LSI algorithm in (19)–(30) is summarized as follows:

1. To initialize, let $k=1$ and

$$\hat{\theta}_0 = \begin{bmatrix} \hat{\boldsymbol{a}}_0 \\ \hat{\boldsymbol{b}}_0 \\ \hat{\boldsymbol{f}}_0 \end{bmatrix}$$

be some nonzero real vector with $\hat{\boldsymbol{a}}_0 \neq 0$, $\hat{\boldsymbol{d}}_0$ be an real vector with $\|\hat{\boldsymbol{d}}_0\| = 1$, and $\hat{v}_0(t-i) = $ random number and form $\hat{\psi}_0(t)$ by (28).
2. Collect the input–output data $\{u(t), y(t), t=1, 2, \ldots, L\}$, and form $\boldsymbol{Y}(L)$ by (22), $\boldsymbol{G}(t)$ by (30), $\boldsymbol{\varphi}(t)$ by (27) and $\boldsymbol{\Phi}(L)$ by (23).
3. Form $\hat{\psi}_k(t)$ by (28), $\hat{\boldsymbol{\Psi}}_k(L)$ by (24), $\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L)$ by (25) and $\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)$ by (21).
4. Compute $\hat{\theta}_k$ by (19), and set

$$\hat{\boldsymbol{a}}_k := \hat{\theta}_k(1:p),$$

$$\hat{\boldsymbol{b}}_k := \hat{\theta}_k(p+1:p+n),$$

$$\hat{\boldsymbol{f}}_k := \hat{\theta}_k(p+n+1:p+n+m).$$

Here, the ":" operation in Matlab is used.
5. Form $\boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)$ by (26).
6. Compute $\hat{\boldsymbol{\vartheta}}_k$ by (20), normalize $\hat{\boldsymbol{d}}_k$ with the first positive element, i.e.,
$$\hat{\boldsymbol{d}}_k = \text{sgn}[\hat{\boldsymbol{\vartheta}}_k(1)] \frac{\hat{\boldsymbol{\vartheta}}_k}{\|\hat{\boldsymbol{\vartheta}}_k\|}.$$

7. Compute $\hat{v}_k(t)$ by (29).
8. Compare $\{\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k, \hat{\boldsymbol{f}}_k, \hat{\boldsymbol{d}}_k\}$ with $\{\hat{\boldsymbol{a}}_{k-1}, \hat{\boldsymbol{b}}_{k-1}, \hat{\boldsymbol{f}}_{k-1}, \hat{\boldsymbol{d}}_{k-1}\}$: if they are sufficiently close, or for some pre-set small $\varepsilon > 0$, if

$$\|\hat{\boldsymbol{a}}_k - \hat{\boldsymbol{a}}_{k-1}\|^2 + \|\hat{\boldsymbol{b}}_k - \hat{\boldsymbol{b}}_{k-1}\|^2 + \|\hat{\boldsymbol{f}}_k - \hat{\boldsymbol{f}}_{k-1}\|^2 + \|\hat{\boldsymbol{d}}_k - \hat{\boldsymbol{d}}_{k-1}\|^2 \leq \varepsilon,$$

then terminate the procedure and obtain the iterative times k and the estimates $\{\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k, \hat{\boldsymbol{f}}_k, \hat{\boldsymbol{d}}_k\}$; otherwise, increase k by 1 and go to step 3.

4. The gradient-based iterative algorithm

This section derives the gradient-based iterative algorithm for the Wiener models. For the optimization problems in (11) and (12), minimizing $J_1(\theta)$ and $J_2(\boldsymbol{\vartheta})$ using the negative gradient search leads to the iterative algorithm of computing $\hat{\theta}_k$ and $\hat{\boldsymbol{\vartheta}}_k$ as follows:

$$\hat{\theta}_k = \hat{\theta}_{k-1} - \frac{\mu_1(k)}{2} \text{grad}[J_1(\hat{\theta}_{k-1})]$$
$$= \hat{\theta}_{k-1} + \mu_1(k)[\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \boldsymbol{\Psi}(L)]^{\text{T}}$$
$$\times \{\boldsymbol{Y}(L) - [\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \boldsymbol{\Psi}(L)]\hat{\theta}_{k-1}\}$$
$$= \hat{\theta}_{k-1} + \mu_1(k)\boldsymbol{\Xi}^{\text{T}}(\hat{\boldsymbol{d}}_{k-1}, L)[\boldsymbol{Y}(L) - \boldsymbol{\Xi}(\hat{\boldsymbol{d}}_{k-1}, L)\hat{\theta}_{k-1}], \quad (31)$$

$$\hat{\boldsymbol{\vartheta}}_k = \hat{\boldsymbol{\vartheta}}_{k-1} - \frac{\mu_2(k)}{2} \text{grad}[J_2(\hat{\boldsymbol{\vartheta}}_{k-1})]$$
$$= \hat{\boldsymbol{\vartheta}}_{k-1} + \mu_2(k)\boldsymbol{\Omega}^{\text{T}}(\hat{\boldsymbol{a}}_k, L)[\boldsymbol{Y}(L) - \boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)\hat{\boldsymbol{d}}_{k-1}$$
$$- \boldsymbol{\Phi}(L)\hat{\boldsymbol{b}}_k - \boldsymbol{\Psi}(L)\hat{\boldsymbol{f}}_k], \quad (32)$$

$$\boldsymbol{\Xi}(\hat{\boldsymbol{d}}_{k-1}, L) = [\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \boldsymbol{\Psi}(L)]. \quad (33)$$

where $\mu_1(k) \geq 0$ and $\mu_2(k) \geq 0$ are the iterative step-sizes or convergence factors to be given later.

However, the similar difficulties arise in that $\psi(t), t=1, 2, \ldots, L$, in $\boldsymbol{\Psi}(L)$ on the right-hand side contain the unmeasured noise terms $v(t-i)$ so it is impossible to compute the estimates $\hat{\theta}_k$ and $\hat{\boldsymbol{\vartheta}}_k$ by (31) and (32). A similar derivation of the W-LSI algorithm is replacing $v(t-i)$ with $\hat{v}_k(t-i)$, $\psi(t)$ with $\hat{\psi}_k(t)$, $\boldsymbol{\Psi}(L)$ with $\hat{\boldsymbol{\Psi}}_k(L)$ and $\boldsymbol{\Xi}(\hat{\boldsymbol{d}}_{k-1}, L)$ with $\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)$, and from (31)–(33) we have

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \mu_1(k)\hat{\boldsymbol{\Xi}}^{\text{T}}(\hat{\boldsymbol{d}}_{k-1}, L)[\boldsymbol{Y}(L) - \hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)\hat{\theta}_{k-1}], \quad (34)$$

$$\hat{\boldsymbol{\vartheta}}_k = \hat{\boldsymbol{\vartheta}}_{k-1} + \mu_2(k)\boldsymbol{\Omega}^{\text{T}}(\hat{\boldsymbol{a}}_k, L)[\boldsymbol{Y}(L) - \boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)\hat{\boldsymbol{d}}_{k-1} - \boldsymbol{\Phi}(L)\hat{\boldsymbol{b}}_k - \hat{\boldsymbol{\Psi}}_k(L)\hat{\boldsymbol{f}}_k], \quad (35)$$

$$\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L) = [\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \hat{\boldsymbol{\Psi}}_k(L)]. \quad (36)$$

Eqs. (34) and (35) can be rewritten as

$$\hat{\theta}_k = [\boldsymbol{I} - \mu_1(k)\hat{\boldsymbol{\Xi}}^{\text{T}}(\hat{\boldsymbol{d}}_{k-1}, L)\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)]\hat{\theta}_{k-1} + \mu_1(k)\hat{\boldsymbol{\Xi}}^{\text{T}}(\hat{\boldsymbol{d}}_{k-1}, L)\boldsymbol{Y}(L), \quad (37)$$

$$\hat{\boldsymbol{\vartheta}}_k = [\boldsymbol{I} - \mu_2(k)\boldsymbol{\Omega}^{\text{T}}(\hat{\boldsymbol{a}}_k, L)\boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)]\hat{\boldsymbol{\vartheta}}_{k-1}$$
$$+ \mu_2(k)\boldsymbol{\Omega}^{\text{T}}(\hat{\boldsymbol{a}}_k, L)[\boldsymbol{Y}(L) - \boldsymbol{\Phi}(L)\hat{\boldsymbol{b}}_k - \hat{\boldsymbol{\Psi}}_k(L)\hat{\boldsymbol{f}}_k]. \quad (38)$$

In order to guarantee the convergence of $\hat{\theta}_k$ and $\hat{\boldsymbol{\vartheta}}_k$, the symmetric matrices $\boldsymbol{I} - \mu_1(k)\hat{\boldsymbol{\Xi}}^{\text{T}}(\hat{\boldsymbol{d}}_{k-1}, L)\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)$ and $\boldsymbol{I} - \mu_2(k)\boldsymbol{\Omega}^{\text{T}}(\hat{\boldsymbol{a}}_k, L)\boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)$ have all eigenvalues inside the unit circle. One conservative choice of $\mu_1(k)$ and $\mu_2(k)$ is to satisfy

$$0 < \mu_1(k) \leq \frac{2}{\lambda_{\max}[\hat{\boldsymbol{\Xi}}^{\text{T}}(\hat{\boldsymbol{d}}_{k-1}, L)\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)]}, \quad (39)$$

$$0 < \mu_2(k) \leq \frac{2}{\lambda_{\max}[\boldsymbol{\Omega}^{\text{T}}(\hat{\boldsymbol{a}}_k, L)\boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)]}. \quad (40)$$

From Eqs. (34)–(36), (39)–(40) and (22)–(30), we can summarize the gradient-based iterative identification algorithm for the Wiener systems (the W-GI algorithm for short) as follows:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \mu_1(k)\hat{\boldsymbol{\Xi}}^{\text{T}}(\hat{\boldsymbol{d}}_{k-1}, L)[\boldsymbol{Y}(L) - \hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)\hat{\theta}_{k-1}], \quad (41)$$

$$\hat{\boldsymbol{\vartheta}}_k = \hat{\boldsymbol{\vartheta}}_{k-1} + \mu_2(k)\boldsymbol{\Omega}^{\text{T}}(\hat{\boldsymbol{a}}_k, L)[\boldsymbol{Y}(L) - \boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)\hat{\boldsymbol{d}}_{k-1} - \boldsymbol{\Phi}(L)\hat{\boldsymbol{b}}_k - \hat{\boldsymbol{\Psi}}_k(L)\hat{\boldsymbol{f}}_k], \quad (42)$$

$$\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L) = [\boldsymbol{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L), \boldsymbol{\Phi}(L), \hat{\boldsymbol{\Psi}}_k(L)], \quad (43)$$

$$0 < \mu_1(k) \leq \frac{2}{\lambda_{\max}[\hat{\boldsymbol{\Xi}}^{\text{T}}(\hat{\boldsymbol{d}}_{k-1}, L)\hat{\boldsymbol{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)]}, \quad (44)$$

$$0 < \mu_2(k) \leq \frac{2}{\lambda_{\max}[\boldsymbol{\Omega}^{\text{T}}(\hat{\boldsymbol{a}}_k, L)\boldsymbol{\Omega}(\hat{\boldsymbol{a}}_k, L)]}, \quad (45)$$

$$\boldsymbol{Y}(L) = [y(1), y(2), \ldots, y(L)]^{\text{T}}, \quad (46)$$

$$\mathbf{\Phi}(L) = [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \ldots, \boldsymbol{\varphi}(L)]^{\mathrm{T}}, \tag{47}$$

$$\hat{\mathbf{\Psi}}_k(L) = [\hat{\boldsymbol{\psi}}_k(1), \hat{\boldsymbol{\psi}}_k(2), \ldots, \hat{\boldsymbol{\psi}}_k(L)]^{\mathrm{T}}, \tag{48}$$

$$\mathbf{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L) = [\boldsymbol{G}(1)\hat{\boldsymbol{d}}_{k-1}, \boldsymbol{G}(2)\hat{\boldsymbol{d}}_{k-1}, \ldots, \boldsymbol{G}(L)\hat{\boldsymbol{d}}_{k-1}]^{\mathrm{T}}, \tag{49}$$

$$\mathbf{\Omega}(\hat{\boldsymbol{a}}_k, L) = [\boldsymbol{G}^{\mathrm{T}}(1)\hat{\boldsymbol{a}}_k, \boldsymbol{G}^{\mathrm{T}}(2)\hat{\boldsymbol{a}}_k, \ldots, \boldsymbol{G}^{\mathrm{T}}(L)\hat{\boldsymbol{a}}_k]^{\mathrm{T}}, \tag{50}$$

$$\boldsymbol{\varphi}(t) = [u(t-1), u(t-2), \ldots, u(t-n)]^{\mathrm{T}}, \quad t = 1, 2, \ldots, L, \tag{51}$$

$$\hat{\boldsymbol{\psi}}_k(t) = [\hat{v}_{k-1}(t-1), \hat{v}_{k-1}(t-2), \ldots, \hat{v}_{k-1}(t-m)]^{\mathrm{T}}, \tag{52}$$

$$\hat{v}_k(t) = y(t) - \hat{\boldsymbol{a}}_{k-1}^{\mathrm{T}} G(t)\hat{\boldsymbol{d}}_{k-1} - \boldsymbol{\varphi}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}_{k-1} + \hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{f}}_{k-1}, \tag{53}$$

$$\boldsymbol{G}(t) = \begin{bmatrix} g_1(y(t-1)) & g_2(y(t-1)) & \cdots & g_q(y(t-1)) \\ g_1(y(t-2)) & g_2(y(t-2)) & \cdots & g_q(y(t-2)) \\ \vdots & \vdots & & \vdots \\ g_1(y(t-p)) & g_2(y(t-p)) & \cdots & g_q(y(t-p)) \end{bmatrix}. \tag{54}$$

The computation process of the W-GI algorithm is summarized as follows:

1. To initialize, let $k=1$ and

$$\hat{\boldsymbol{\theta}}_0 = \begin{bmatrix} \hat{\boldsymbol{a}}_0 \\ \hat{\boldsymbol{b}}_0 \\ \hat{\boldsymbol{f}}_0 \end{bmatrix}$$

be some nonzero real vector with $\hat{\boldsymbol{a}}_0 \neq 0$, and $\hat{\boldsymbol{d}}_0$ be an real vector with $\|\hat{\boldsymbol{d}}_0\| = 1$, and $\hat{v}_0(t-i) = 0$ and form $\hat{\boldsymbol{\psi}}_0(t)$ by (52).
2. Collect the input–output data $\{u(t), y(t), t=1, 2, \ldots, L\}$, and form $\boldsymbol{Y}(L)$ by (46), $\boldsymbol{G}(t)$ by (54), $\boldsymbol{\varphi}(t)$ by (51) and $\mathbf{\Phi}(L)$ by (47).
3. Form $\hat{\boldsymbol{\psi}}_k(t)$ by (52), $\hat{\mathbf{\Psi}}_k(L)$ by (48), $\mathbf{\Upsilon}(\hat{\boldsymbol{d}}_{k-1}, L)$ by (49) and $\hat{\mathbf{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)$ by (43).
4. Choose $\mu_1(t)$ and $\mu_2(t)$ according to (44) and (45), compute $\hat{\boldsymbol{\theta}}_k$ by (41), and set

$$\hat{\boldsymbol{a}}_k := \hat{\boldsymbol{\theta}}_k(1:p),$$

$$\hat{\boldsymbol{b}}_k := \hat{\boldsymbol{\theta}}_k(p+1:p+n),$$

$$\hat{\boldsymbol{f}}_k := \hat{\boldsymbol{\theta}}_k(p+n+1:p+n+m).$$

5. Form $\mathbf{\Omega}(\hat{\boldsymbol{a}}_k, L)$ by (50).
6. Compute $\hat{\boldsymbol{\vartheta}}_k$ by (42), and normalize $\hat{\boldsymbol{d}}_k$ with the first positive element, i.e.,
$$\hat{\boldsymbol{d}}_k = \mathrm{sgn}[\hat{\boldsymbol{\vartheta}}_k(1)]\frac{\hat{\boldsymbol{\vartheta}}_k}{\|\hat{\boldsymbol{\vartheta}}_k\|}.$$

7. Compute $\hat{v}_k(t)$ by (53).
8. Compare $\{\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k, \hat{\boldsymbol{f}}_k, \hat{\boldsymbol{d}}_k\}$ with $\{\hat{\boldsymbol{a}}_{k-1}, \hat{\boldsymbol{b}}_{k-1}, \hat{\boldsymbol{f}}_{k-1}, \hat{\boldsymbol{d}}_{k-1}\}$: if they are sufficiently close, or for some pre-set small $\varepsilon > 0$, if

$$\|\hat{\boldsymbol{a}}_k - \hat{\boldsymbol{a}}_{k-1}\|^2 + \|\hat{\boldsymbol{b}}_k - \hat{\boldsymbol{b}}_{k-1}\|^2 + \|\hat{\boldsymbol{f}}_k - \hat{\boldsymbol{f}}_{k-1}\|^2 + \|\hat{\boldsymbol{d}}_k - \hat{\boldsymbol{d}}_{k-1}\|^2 \leq \varepsilon,$$

then terminate the procedure and obtain the iterative times k and the estimates $\{\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_k, \hat{\boldsymbol{f}}_k, \hat{\boldsymbol{d}}_k\}$; otherwise, increment k by 1 and go to step 3.

In the above W-LSI and W-GI algorithms, the noise $v(t)$ is computed by the iterative technique, which is just like in the identification for Hammerstein nonlinear ARMAX systems in [1], avoiding the computation of the covariance matrix of the noise. Moreover, the estimate of $v(t)$ converges its true value as long as the parameter estimates converge their true values.

For CBDS-CDMA communication systems, the jamming interferences or data-missing cases often arise. In the cases, the auxiliary model based identification method can be applied [34].

5. Example

Consider the following nonlinear system with colored noise:

$$y(t) = \sum_{i=1}^{2} a_i[d_1 g(y(t-i)) + d_2 g_2(y(t-i)) + d_3 g_3(y(t-i))]$$
$$+ \sum_{j=1}^{2} b_j u(t-j) + fv(t-1) + v(t),$$

$$g_1(y(t-i)) = y(t-i),$$

$$g_2(y(t-i)) = y^2(t-i),$$

$$g_3(y(t-i)) = y^3(t-i),$$

$$\boldsymbol{\theta} = [a_1, a_2, b_1, b_2, f]^{\mathrm{T}} = [0.25, 0.28, -0.30, 1.00, 0.05]^{\mathrm{T}},$$

$$\boldsymbol{\vartheta} = [d_1, d_2, d_3]^{\mathrm{T}} = [0.80, -0.50, -0.3317]^{\mathrm{T}},$$

$$\Theta = [\boldsymbol{\theta}^{\mathrm{T}}, \boldsymbol{\vartheta}^{\mathrm{T}}]^{\mathrm{T}} = [0.25, 0.28, -0.30, 1.00, 0.05, 0.80, -0.50, -0.3317]^{\mathrm{T}}.$$

In simulation, the input $\{u(t)\}$ is taken as a persistent excitation signal sequence with zero mean and unit variance, and $\{v(t)\}$ as a white noise sequence with zero mean and variance $\sigma^2 = 0.20^2$.

Apply the proposed W-LSI and W-GI algorithms to estimate the parameters of this system, the parameter estimates and their errors with different data length L are shown in Tables 1–4 and the parameter estimation errors $\delta := \|\hat{\Theta}_k - \Theta\|/\|\Theta\|$ versus k are shown in Fig. 1, where the convergence factors of the W-GI algorithm are taken to be

$$\mu_1(k) = \frac{1.5}{\lambda_{\max}[\hat{\mathbf{\Xi}}^{\mathrm{T}}(\hat{\boldsymbol{d}}_{k-1}, L)\hat{\mathbf{\Xi}}(\hat{\boldsymbol{d}}_{k-1}, L)]},$$

$$\mu_2(k) = \frac{1.5}{\lambda_{\max}[\mathbf{\Omega}^{\mathrm{T}}(\hat{\boldsymbol{a}}_k, L)\mathbf{\Omega}(\hat{\boldsymbol{a}}_k, L)]}.$$

From Tables 1–4 and Fig. 1, we can draw the following conclusions.

- The parameter estimation errors given by the W-LSI and W-GI algorithms become small as the iterations increase.
- The parameter estimation errors given by the W-LSI and W-GI algorithms become small with the data length L increasing.
- The W-LSI algorithm has faster convergence rates than the W-GI algorithm. The W-LSI algorithm can generate highly accurate parameter estimates after only several iterations.

Table 1
The W-LSI parameter estimates and errors ($L=1000$).

k	a_1	a_2	b_1	b_2	f	d_1	d_2	d_3	δ (%)
1	0.16830	0.20117	−0.29866	0.96788	0.16198	0.47697	−0.73778	−0.47768	30.56569
2	0.25089	0.28526	−0.29876	1.00102	0.01518	0.78248	−0.51886	−0.34426	3.03989
3	0.25166	0.28234	−0.29944	1.00384	0.07784	0.78730	−0.51664	−0.33652	2.37482
4	0.25168	0.28204	−0.29949	1.00402	0.08271	0.78744	−0.51671	−0.33609	2.63431
5	0.25168	0.28201	−0.29948	1.00403	0.08249	0.78744	−0.51673	−0.33606	2.62186
6	0.25168	0.28201	−0.29948	1.00403	0.08244	0.78744	−0.51672	−0.33605	2.61901
7	0.25168	0.28201	−0.29948	1.00403	0.08245	0.78745	−0.51672	−0.33605	2.61919
8	0.25168	0.28201	−0.29948	1.00403	0.08245	0.78745	−0.51672	−0.33605	2.61922
9	0.25168	0.28201	−0.29948	1.00403	0.08245	0.78745	−0.51672	−0.33605	2.61921
10	0.25168	0.28201	−0.29948	1.00403	0.08245	0.78745	−0.51672	−0.33605	2.61921
True values	0.25000	0.28000	−0.30000	1.00000	0.05000	0.80000	−0.50000	−0.33170	

Table 2
The W-LSI parameter estimates and errors ($L=2000$).

k	a_1	a_2	b_1	b_2	f	d_1	d_2	d_3	δ (%)
1	0.16929	0.19889	−0.30377	0.97414	0.15759	0.47862	−0.73854	−0.47485	30.37233
2	0.25358	0.27905	−0.30442	1.00223	0.02044	0.79027	−0.51117	−0.33790	2.28847
3	0.25345	0.27533	−0.30459	1.00442	0.05028	0.79264	−0.51167	−0.33154	1.08855
4	0.25349	0.27513	−0.30461	1.00453	0.05105	0.79291	−0.51150	−0.33115	1.08224
5	0.25349	0.27512	−0.30461	1.00454	0.05098	0.79293	−0.51148	−0.33112	1.08102
6	0.25349	0.27512	−0.30461	1.00454	0.05098	0.79293	−0.51148	−0.33112	1.08089
7	0.25349	0.27512	−0.30461	1.00454	0.05098	0.79294	−0.51148	−0.33112	1.08089
8	0.25349	0.27512	−0.30461	1.00454	0.05098	0.79294	−0.51148	−0.33112	1.08089
9	0.25349	0.27512	−0.30461	1.00454	0.05098	0.79294	−0.51148	−0.33112	1.08089
10	0.25349	0.27512	−0.30461	1.00454	0.05098	0.79294	−0.51148	−0.33112	1.08089
True values	0.25000	0.28000	−0.30000	1.00000	0.05000	0.80000	−0.50000	−0.33170	

Table 3
The W-GI parameter estimates and errors ($L=1000$).

k	a_1	a_2	b_1	b_2	f	d_1	d_2	d_3	δ (%)
1	−0.33294	0.24567	−0.03180	0.22133	−0.01676	0.18630	0.38146	0.90542	128.84602
5	0.04178	0.06589	−0.28319	0.88955	0.05812	0.08379	−0.89501	−0.43810	59.17438
10	0.14025	0.16213	−0.30083	0.99479	0.06761	0.25766	−0.90546	−0.33727	46.59230
50	0.23741	0.26880	−0.29954	1.00241	0.07227	0.75822	−0.55458	−0.34286	5.02301
100	0.25144	0.28179	−0.29948	1.00400	0.08147	0.78698	−0.51735	−0.33617	2.59360
150	0.25168	0.28201	−0.29948	1.00403	0.08237	0.78744	−0.51673	−0.33605	2.61527
200	0.25168	0.28201	−0.29948	1.00403	0.08244	0.78745	−0.51672	−0.33605	2.61887
True values	0.25000	0.28000	−0.30000	1.00000	0.05000	0.80000	−0.50000	−0.33170	

Table 4
The W-GI parameter estimates and errors ($L=2000$).

| k	a_1	a_2	b_1	b_2	f	d_1	d_2	d_3	δ (%)
1	−0.33819	0.25057	−0.03745	0.25259	−0.01900	0.17776	0.45542	0.87235	129.21231
5	0.16026	0.12186	−0.27999	0.93550	0.00296	0.14090	−0.96159	−0.23555	55.84611
10	0.14535	0.15828	−0.30302	1.00198	0.00993	0.31689	−0.89422	−0.31614	43.18056
50	0.24552	0.26794	−0.30454	1.00384	0.04343	0.77680	−0.53342	−0.33472	2.92301
100	0.25342	0.27505	−0.30460	1.00454	0.05061	0.79279	−0.51169	−0.33115	1.09366
150	0.25349	0.27512	−0.30461	1.00454	0.05096	0.79293	−0.51148	−0.33112	1.08091
200	0.25349	0.27512	−0.30461	1.00454	0.05098	0.79294	−0.51148	−0.33112	1.08088
True values	0.25000	0.28000	−0.30000	1.00000	0.05000	0.80000	−0.50000	−0.33170	
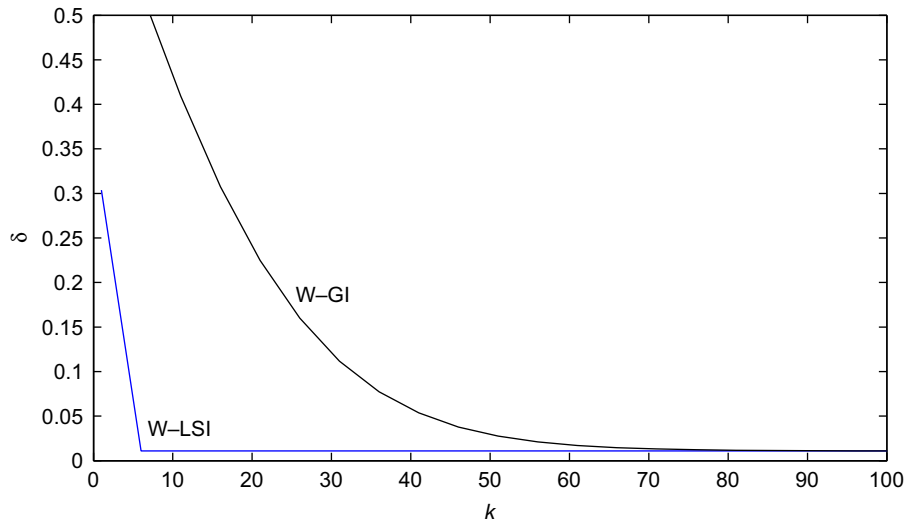
Fig. 1. The parameter estimation errors δ versus k (L=2000).

6. Conclusions

A least squares-based and a gradient-based iterative algorithms are developed for Wiener nonlinear systems using the hierarchical identification principle. The proposed two iterative algorithms can give a satisfactory identification accuracy and the least squares-based iterative algorithm has faster convergence rates than the gradient-based iterative algorithm and requires computing the matrix inversion. Although the algorithms are presented for the Wiener models, the basic idea can also be extended to identify Hammerstein–Wiener models.

The proposed least squares-based and gradient-based iterative algorithms are similar to maximum likelihood methods to some extent and can be applied to multirate nonlinear systems or non-uniformly sampled systems [35–39]. The data filtering-based identification method in [40] can be extended to such Wiener nonlinear systems. In general, the parameter estimation errors of the identification algorithms become small as the data length increases. This paper uses the example to verify the convergence results of the proposed iterative algorithms for nonlinear systems. The convergence analysis of the iterative algorithms is complex and requires further study. The proposed iterative algorithm can reduce to identify Hammerstein systems [41].

References

[1] F. Ding, T. Chen, Identification of Hammerstein nonlinear ARMAX systems, Automatica 41 (9) (2005) 1479–1489.
[2] F. Ding, Y. Shi, T. Chen, Auxiliary model based least-squares identification methods for Hammerstein output-error systems, Systems & Control Letters 56 (5) (2007) 373–380.
[3] I.J. Umoh, T. Ogunfunmi, An affine projection-based algorithm for identification of nonlinear Hammerstein systems, Signal Processing 90 (6) (2010) 2020–2030.
[4] D.Q. Wang, Y.Y. Chu, F. Ding, Auxiliary model-based RELS and MI-ELS algorithms for Hammerstein OEMA systems, Computers & Mathematics with Applications 59 (9) (2010) 3092–3098.

[5] D.Q. Wang, Y.Y. Chu, G.W. Yang, F. Ding, Auxiliary model-based recursive generalized least squares parameter estimation for Hammerstein OEAR systems, Mathematical and Computer Modelling 52 (1–2) (2010) 309–317.
[6] J. Vörös, Modeling and identification of Wiener systems with two-segment nonlinearities, IEEE Transactions on Control Systems Technology 11 (2) (2003) 253–257.
[7] J. Vörös, Parameter identification of Wiener systems with multi-segment piecewise-linear nonlinearities, Systems & Control Letters 56 (2) (2007) 99–105.
[8] L.A. Aguirre, M.C.S. Coelho, M.V. Corrêa, On the interpretation and practice of dynamical difference between Hammerstein and Wiener models, IEE Prooceedings of Control Theory and Applications 152 (4) (2005) 349–356.
[9] X.L. Hu, H.F. Chen, Strong consistence of recursive identification for Wiener systems, Automatica 41 (11) (2005) 1905–1916.
[10] M. Kozek, S. Sinanović, Identification of Wiener models using optimal local linear models, Simulation Modelling Practice and Theory 16 (8) (2008) 1055–1066.
[11] J.L. Figueroa, S.I. Biagiola, O.E. Agamennoni, An approach for identification of uncertain Wiener systems, Mathematical and Computer Modelling 48 (1–2) (2008) 305–315.
[12] A. Hagenblad, L. Ljung, A. Wills, Maximum likelihood identification of Wiener models, Automatica 44 (11) (2008) 2697–2705.
[13] F. Ding, P.X. Liu, G. Liu, Gradient based and least-squares based iterative identification methods for OE and OEMA systems, Digital Signal Processing 20 (3) (2010) 664–677.
[14] Y.J. Liu, D.Q. Wang, F. Ding, Least-squares based iterative algorithms for identifying Box-Jenkins models with finite measurement data, Digital Signal Processing 20 (5) (2010) 1458–1467.
[15] D.Q. Wang, G.W. Yang, R.F. Ding, Gradient-based iterative parameter estimation for Box-Jenkins systems, Computers & Mathematics with Applications 60 (5) (2010) 1200–1208.
[16] G. Kapetanios, A note on an iterative least-squares estimation method for ARMA and VARMA models, Economics Letters 79 (3) (2003) 305–312.
[17] E.W. Bai, Y. Liu, Least squares solutions of bilinear equations, Systems & Control Letters 55 (6) (2006) 466–472.
[18] F. Ding, T. Chen, Hierarchical gradient-based identification of multi-variable discrete-time systems, Automatica 41 (2) (2005) 315–325.
[19] F. Ding, T. Chen, Hierarchical least squares identification methods for multivariable systems, IEEE Transactions on Automatic Control 50 (3) (2005) 397–402.
[20] F. Ding, T. Chen, Hierarchical identification of lifted state-space models for general dual-rate systems, IEEE Transactions on Circuits and Systems—I: Regular Papers 52 (6) (2005) 1179–1187.
[21] D.Q. Wang, Y.Y. Chu, F. Ding, Identification methods for Wiener nonlinear systems based on the least squares and gradient iterations, in: 48th IEEE Conference on Decision and Control and 28th Chinese

Control Conference, December 16–18, 2009, Shanghai, China, pp. 3632–3636.

[22] F. Ding, T. Chen, Performance analysis of multi-innovation gradient type identification methods, Automatica 43 (1) (2007) 1–14.

[23] F. Ding, Several multi-innovation identification methods, Digital Signal Processing 20 (4) (2010) 1027–1039.

[24] D.Q. Wang, F. Ding, Performance analysis of the auxiliary models based multi-innovation stochastic gradient estimation algorithm for output error systems, Digital Signal Processing 20 (3) (2010) 750–762.

[25] F. Ding, P.X. Liu, G. Liu, Auxiliary model based multi-innovation extended stochastic gradient parameter estimation with colored measurement noises, Signal Processing 89 (10) (2009) 1883–1890.

[26] L.L. Han, F. Ding, Multi-innovation stochastic gradient algorithms for multi-input multi-output systems, Digital Signal Processing 19 (4) (2009) 545–554.

[27] J.B. Zhang, F. Ding, Y. Shi, Self-tuning control based on multi-innovation stochastic gradient parameter estimation, Systems & Control Letters 58 (1) (2009) 69–75.

[28] F. Ding, P. X Liu, G. Liu, Multi-innovation least squares identification for linear and pseudo-linear regression models, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 40 (3) (2010) 767–778.

[29] L. Xie, Y.J. Liu, H.Z. Yang, F. Ding, Modeling and identification for non-uniformly periodically sampled-data systems, IET Control Theory & Applications 4 (5) (2010) 784–794.

[30] E.W. Bai, An optimal two-stage identification algorithm for Hammerstein–Wiener nonlinear systems, Automatica 34 (3) (1998) 333–338.

[31] D.Q. Wang, F. Ding, Extended stochastic gradient identification algorithms for Hammerstein–Wiener ARMXA Systems, Computers and Mathematics with Applications 56 (12) (2008) 3157–3164.

[32] Y. Liu, E.W. Bai, Iterative identification of Hammerstein systems, Automatica 43 (2) (2007) 346–354.

[33] V. Cerone, D. Regruto, Parameter bounds for discrete-time Hammerstein models with bounded output errors, IEEE Transactions on Automatic Control 48 (10) (2003) 1855–1860.

[34] F. Ding, J. Ding, Least squares parameter estimation with irregularly missing data, International Journal of Adaptive Control and Signal Processing 24 (7) (2010) 540–553.

[35] F. Ding, L. Qiu, T. Chen, Reconstruction of continuous-time systems from their non-uniformly sampled discrete-time systems, Automatica 45 (2) (2009) 324–332.

[36] F. Ding, G. Liu, X.P. Liu, Partially coupled stochastic gradient identification methods for non-uniformly sampled systems, IEEE Transaction on Automatic Control 55 (8) (2010) 1976–1981.

[37] Y.J. Liu, L. Xie, F. Ding, An auxiliary model based recursive least squares parameter estimation algorithm for non-uniformly sampled multirate systems, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 223 (4) (2009) 445–454.

[38] F. Ding, P.X. Liu, H.Z. Yang, Parameter identification and intersample output estimation for dual-rate systems, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 38 (4) (2008) 966–975.

[39] J. Ding, Y. Shi, H.G. Wang, F. Ding, A modified stochastic gradient based parameter estimation algorithm for dual-rate sampled-data systems, Digital Signal Processing 20 (4) (2010) 1238–1249.

[40] D.Q. Wang, F. Ding, Input–output data filtering based recursive least squares identification for CARARMA systems, Digital Signal Processing 20 (4) (2010) 991–999.

[41] F. Ding, P.X. Liu, G. Liu, Identification methods for Hammerstein nonlinear systems, Digital Signal Processing 21 (2) (2011), doi:10.1016/j.dsp.2010.06.006.