Hindawi
Complexity
Volume 2018, Article ID 7353171, 8 pages
https://doi.org/10.1155/2018/7353171

Research Article

Particle Swarm Optimization Iterative Identification Algorithm and Gradient Iterative Identification Algorithm for Wiener Systems with Colored Noise

Junhong Li and Xiao Li

School of Electrical Engineering, Nantong University, Nantong 226019, China

Correspondence should be addressed to Junhong Li; missjunhong@163.com

Received 16 April 2018; Accepted 23 May 2018; Published 24 July 2018

Academic Editor: Jing Na

This paper considers the parameter identification of Wiener systems with colored noise. The difficulty in the identification is that the model is nonlinear and the intermediate variable cannot be measured. Particle swarm optimization is an artificial intelligence evolutionary method and is effective in solving nonlinear optimization problem. In this paper, we obtain the identification model of the Wiener system and then transfer the parameter identification problem into an optimization problem. Then, we derive a particle swarm optimization iterative (PSOI) identification algorithm to identify the unknown parameter of the Wiener system. Furthermore, a gradient iterative identification algorithm is proposed to compare with the particle swarm optimization iterative algorithm. Numerical simulation is carried out to evaluate the performance of the PSOI algorithm and the gradient iterative algorithm. The simulation results indicate that the proposed algorithms are effective and the PSOI algorithm can achieve better performance over the gradient iterative algorithm.

1. Introduction

Almost all practical systems are nonlinear [1–3]. Many identification methods have been developed for linear systems [4, 5], bilinear systems [6–8], and nonlinear systems [9]. The Wiener models are a typical class of nonlinear systems and are widely used in industrial production process [10, 11]. The Wiener nonlinear system consists of a dynamic linear subsystem and a static nonlinear subsystem and has the characteristics of complex structure between subsystems [12, 13]. One of the difficulties in identifying Wiener nonlinear model parameters is that the intermediate variable (the output of the linear subsystem) cannot be measured, and the identification issues for Wiener systems have attracted great attention [14].

The iterative identification method is generally used to identify the system with unknown item in the model information vector [15–17]. The basic idea of iterative identification is to estimate the unknown items in the information vector by using the iterative parameter estimation of the previous step [18, 19]. The iterative identification method is an important branch of system identification, which can be realized by using gradient search, least squares principle, and Newton optimization [20–22].

The particle swarm optimization algorithm is an evolutionary computing technique which is based on the simulation of birds' flock [23, 24]. The basic idea of particle swarm optimization algorithm is to find the optimal solution through collaboration and information sharing among individuals in the group [25]. This algorithm has attracted the attention of academia with the advantages of easy implementation, high precision, and fast convergence [26]. Compared with the conventional optimization methods, it has excellent optimized performances and characteristics [27]. The particle swarm optimization algorithm has been widely used in function optimization, system identification, and fuzzy control [28–30]. Recently, Chen and Wang proposed a stochastic gradient algorithm and a particle swarm optimization algorithm to estimate all the unknown parameters of the Hammerstein system

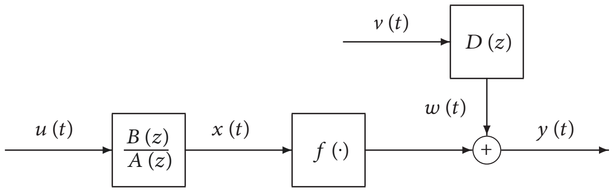

Figure 1: The Wiener system model.

[31]. In this paper, we use the particle swarm optimization algorithm and the gradient iterative algorithm to identify the unknown parameters of the Wiener systems with colored noise.

The rest of the paper is organized as follows. Section 2 gives the system description for the Wiener model. Section 3 gives the particle swarm optimization algorithm for Wiener nonlinear systems. Section 4 derives a gradient iterative algorithm for the discussed system. Section 5 provides an example for illustrating the results in this paper. Finally, some conclusions are given in Section 6.

2. System Description

Consider the Wiener system shown in Figure 1 with the following expressions:

$$\begin{aligned} y(t) &:= f(x(t)) + D(z)v(t), \\ x(t) &:= \frac{B(z)}{A(z)}u(t), \end{aligned} \tag{1}$$

where $A(z)$, $B(z)$, and $D(z)$ are polynomials in the shift operator $z^{-1}[z^{-1}y(t) = y(t-1)]$ with

$$\begin{aligned} A(z) &:= 1 + a_1 z^{-1} + a_2 z^{-1} + \cdots + a_{n_a} z^{-n_a}, \\ B(z) &:= b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b}, \\ D(z) &:= 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_{n_d} z^{-n_d}. \end{aligned} \tag{2}$$

Assume that the degrees n_a, n_b, and n_d are known and $y(t) = 0$, $u(t) = 0$, and $v(t) = 0$ for $t \leq 0$.

Define the linear subsystem output $x(t)$ as

$$x(t) := \frac{B(z)}{A(z)}u(t), \tag{3}$$

and the noise model output $w(t)$ as

$$w(t) := D(z)v(t). \tag{4}$$

The static nonlinear block is a nonlinear function

$$f(x(t)) := \gamma_1 f_1(x(t)) + \gamma_2 f_2(x(t)) + \cdots + \gamma_{n_\gamma} f_{n_\gamma}(x(t)), \tag{5}$$

where the basis $g := (f_1, f_2, \ldots, f_{n_\gamma})$ are known nonlinear functions of $x(t)$, the unknown parameters γ_i are the coefficients of the nonlinear functions and assume that the degree n_γ is known. Without loss of generality, let the first coefficient of nonlinear block γ_1 be unity and rewrite the $f(x(t))$ as

$$f(x(t)) = x(t) + \gamma_2 x^2(t) + \cdots + \gamma_{n_\gamma} x^{n_\gamma}(t). \tag{6}$$

In the above equations, $u(t)$ and $y(t)$ are the system input and output, respectively, and $v(t)$ is a Gaussian distributed white noise with zero mean and variance σ^2. From (3), we have

$$\begin{aligned} x(t) &= [1 - A(z)]x(t) + B(z)u(t) \\ &= -a_1 x(t-1) - \cdots - a_{n_a} x(t-n_a) \\ &\quad + b_1 u(t-1) + \cdots + b_{n_b} u(t-n_b) \\ &= \varphi_1^{\mathrm{T}}(t)\boldsymbol{\theta}_1, \end{aligned} \tag{7}$$

where

$$\begin{aligned} \varphi_1(t) &:= [-x(t-1), \ldots, -x(t-n_a), u(t-1), \ldots, u(t-n_b)]^{\mathrm{T}} \\ &\in \mathbb{R}^{n_a+n_b}, \\ \boldsymbol{\theta}_1 &:= \left[a_1, a_2, \ldots, a_{n_a}, b_1, b_2, \ldots, b_{n_b}\right]^{\mathrm{T}} \\ &\in \mathbb{R}^{n_a+n_b}. \end{aligned} \tag{8}$$

From (4), we can obtain

$$\begin{aligned} w(t) &= D(z)v(t) \\ &= d_1 v(t-1) + d_2 v(t-2) + \cdots + d_{n_d} v(t-n_d) \\ &= \varphi_d^{\mathrm{T}}(t)\boldsymbol{\theta}_d, \end{aligned} \tag{9}$$

where

$$\begin{aligned} \varphi_d(t) &:= [v(t-1), v(t-2), \ldots, v(t-n_d)]^{\mathrm{T}} \in \mathbb{R}^{n_d}, \\ \boldsymbol{\theta}_d &:= \left[d_1, d_2, \ldots, d_{n_d}\right]^{\mathrm{T}} \in \mathbb{R}^{n_d}. \end{aligned} \tag{10}$$

Thus, the Wiener nonlinear system model can be written as follows:

$$\begin{aligned} y(t) &= f(x(t)) + w(t) \\ &= x(t) + \gamma_2 x^2(t) + \cdots + \gamma_{n_\gamma} x^{n_\gamma}(t) + D(z)v(t) \\ &= \varphi_1^{\mathrm{T}}(t)\boldsymbol{\theta}_1 + \varphi_2^{\mathrm{T}}(t)\boldsymbol{\theta}_2 + \varphi_d^{\mathrm{T}}(t)\boldsymbol{\theta}_d + v(t) \\ &= \varphi^{\mathrm{T}}(t)\boldsymbol{\theta} + v(t), \end{aligned} \tag{11}$$

where

$$\begin{aligned} \varphi_2(t) &:= \left[x^2(t), x^3(t), \ldots, x^{n_r}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{n_\gamma-1}, \\ \boldsymbol{\theta}_2 &:= \left[\gamma_2, \gamma_3, \ldots, \gamma_{n_\gamma}\right]^{\mathrm{T}} \in \mathbb{R}^{n_\gamma-1}, \\ \varphi(t) &:= \left[\varphi_1^{\mathrm{T}}(t), \varphi_2^{\mathrm{T}}(t), \varphi_d^{\mathrm{T}}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{n}, \\ \boldsymbol{\theta} &:= \left[\theta_1^{\mathrm{T}}, \theta_2^{\mathrm{T}}, \theta_d^{T}\right]^{\mathrm{T}} \in \mathbb{R}^{n}, \\ n &:= n_a + n_b + n_\gamma - 1 + n_d. \end{aligned} \tag{12}$$

3. The Particle Swarm Optimization Algorithm

With the development of optimization theory, some new intelligent algorithms have been proposed to solve the problem of traditional system identification, such as the genetic algorithm [32], the ant colony algorithm [33], and the particle swarm algorithm [34, 35], these algorithms enrich the system identification technology. Particle swarm optimization algorithm is a nature-inspired evolutionary algorithm, and it has been successful in solving a wide range of real-value optimization problems [36]. In the following, the particle swarm optimization algorithm is used to identify the unknown parameters of the Wiener nonlinear systems.

Suppose that the search space is n-dimensional and a particle swarm consists of M particles.

Define the information vector as

$$\begin{aligned}\boldsymbol{\varphi}(t) := \big[&-x(t-1), \dots, -x(t-n_a), u(t-1), \dots, \\ &u(t-n_b), x^2(t), x^3(t), \dots, x^{n_r}(t), \\ &v(t-1), v(t-2), \dots, v(t-n_d)\big]^{\mathrm{T}} \in \mathbb{R}^n.\end{aligned} \tag{13}$$

Let p represent the data length. Define the stacked output vector $\mathbf{Y}(p)$ and the stacked information matrix $\boldsymbol{\Phi}(p)$ as

$$\begin{aligned}\mathbf{Y}(p) &:= [y(p), y(p-1), \dots, y(1)]^{\mathrm{T}} \in \mathbb{R}^p, \\ \boldsymbol{\Phi}(p) &:= [\varphi(p), \varphi(p-1), \dots, \varphi(1)] \in \mathbb{R}^{n\times p}.\end{aligned} \tag{14}$$

Define the independent position of each particle $\widehat{\boldsymbol{\theta}}_i$ and the independent velocity $\mathbf{Q}_i$ as follows:

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_i &:= \big[\widehat{a}_{1i}, \widehat{a}_{2i}, \dots, \widehat{a}_{n_a i}, \widehat{b}_{1i}, \widehat{b}_{2i}, \dots, \widehat{b}_{n_b i}, \widehat{\gamma}_{2i}, \\ &\quad \widehat{\gamma}_{3i}, \dots, \widehat{\gamma}_{n_\gamma i}, d_{1i}, d_{2i}, \dots, \widehat{d}_{n_d i}\big]^{\mathrm{T}} \in \mathbb{R}^n, \\ \mathbf{Q}_i &:= [q_{1i}, q_{2i}, \dots, q_{ni}]^{\mathrm{T}} \in \mathbb{R}^n, \quad i = 1, 2, \dots, M.\end{aligned} \tag{15}$$

Let $\widehat{\boldsymbol{\theta}}(k)$ denote the estimates of $\widehat{\boldsymbol{\theta}}$ at iteration $k (k = 1, 2, 3, \dots)$. Define $\widehat{\boldsymbol{\theta}}_{ih}(k)$ as the best position of each particle at iteration k

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{ih}(k) := \big[&\widehat{a}_{1ih}(k), \dots, \widehat{a}_{n_a ih}(k), \widehat{b}_{1ih}(k), \dots, \widehat{b}_{n_b ih}(k), \\ &\widehat{\gamma}_{2ih}(k), \dots, \widehat{\gamma}_{n_\gamma ih}(k), \widehat{d}_{1ih}(k), \dots, \widehat{d}_{n_d ih}(k)\big]^{\mathrm{T}} \in \mathbb{R}^n.\end{aligned} \tag{16}$$

Let $\widehat{\varphi}_{1,k}(t)$ and $\widehat{\varphi}_k(t)$ denote the estimates of $\widehat{\varphi}_1(t)$ and $\widehat{\varphi}(t)$ at iteration k

$$\begin{aligned}\widehat{\varphi}_{1,k}(t) &:= \big[-\widehat{x}_{k-1}(t-1), \dots, -\widehat{x}_{k-1}(t-n_a), \\ &\quad u(t-1), \dots, u(t-n_b)\big]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_b}, \\ \widehat{\varphi}_k(t) &:= \big[-\widehat{x}_{k-1}(t-1), \dots, -\widehat{x}_{k-1}(t-n_a), \\ &\quad u(t-1), \dots, u(t-n_b), \widehat{x}_{k-1}^2(t), \\ &\quad \widehat{x}_{k-1}^3(t), \dots, \widehat{x}_{k-1}^{n_r}(t), \widehat{v}_{k-1}(t-1), \\ &\quad \widehat{v}_{k-1}(t-2), \dots, \widehat{v}_{k-1}(t-n_d)\big]^{\mathrm{T}} \in \mathbb{R}^n.\end{aligned} \tag{17}$$

Then, the estimates $\widehat{\boldsymbol{\Phi}}_k(p)$ can be obtained as follows:

$$\widehat{\boldsymbol{\Phi}}_k(p) = [\widehat{\varphi}_k(p), \widehat{\varphi}_k(p-1), \dots, \widehat{\varphi}_k(1)]. \tag{18}$$

According to the basic principle of the particle swarm algorithm, the best position of each particle $\widehat{\boldsymbol{\theta}}_{ih}(k)$ satisfies the following cost function:

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{ih}(k) &:= \arg\min \\ &\left(\left\|\mathbf{Y}(p) - \widehat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(p)\widehat{\boldsymbol{\theta}}_i(k)\right\|, \left\|\mathbf{Y}(p) - \widehat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(p)\widehat{\boldsymbol{\theta}}_{ih}(k-1)\right\|\right).\end{aligned} \tag{19}$$

Let $\widehat{\boldsymbol{\theta}}_g(k)$ denote the global best position of all the particles

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_g(k) := \big[&\widehat{a}_{1g}(k), \dots, \widehat{a}_{n_a g}(k), \widehat{b}_{1g}(k), \dots, \widehat{b}_{n_b g}(k), \\ &\widehat{\gamma}_{2g}(k), \dots, \widehat{\gamma}_{n_\gamma g}(k), \widehat{d}_{1g}(k), \dots, \widehat{d}_{n_d g}(k)\big]^{\mathrm{T}} \in \mathbb{R}^n,\end{aligned} \tag{20}$$

where $\widehat{\boldsymbol{\theta}}_g(k)$ satisfies

$$\widehat{\boldsymbol{\theta}}_g(k) = \arg\min \left(\left\|\mathbf{Y}(p) - \widehat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(p)\widehat{\theta}_{ih}(k)\right\|\right). \tag{21}$$

Define $\widehat{\boldsymbol{\theta}}_{g1}(k) := [\widehat{a}_{1g}(k), \dots, \widehat{a}_{n_a g}(k), \widehat{b}_{1g}(k), \dots, \widehat{b}_{n_b g}(k)]^{\mathrm{T}}$. According to (7), we can obtain the estimation of $\widehat{x}_k(t)$

$$\widehat{x}_k(t) = \widehat{\boldsymbol{\varphi}}_{1,k}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}_{g1}(k). \tag{22}$$

According to the principle of particle swarm optimization, each particle goes to a new position and a new velocity at iteration $k+1$ as follows:

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_i(k+1) &= \widehat{\boldsymbol{\theta}}_i(k) + \widehat{\mathbf{Q}}_i(k+1), \\ \widehat{\mathbf{Q}}_i(k+1) &= \beta\widehat{\mathbf{Q}}_i(k) + \xi_1\zeta_1\left(\widehat{\boldsymbol{\theta}}_{ih}(k) - \widehat{\boldsymbol{\theta}}_i(k)\right) \\ &\quad + \xi_2\zeta_2\left(\widehat{\boldsymbol{\theta}}_g(k) - \widehat{\boldsymbol{\theta}}_i(k)\right).\end{aligned} \tag{23}$$

Replacing $\varphi(t)$ and $\boldsymbol{\theta}$ with $\widehat{\varphi}_k(t)$ and $\widehat{\boldsymbol{\theta}}_g(k)$ in (11), we can obtain the estimate $\widehat{v}_k(t) = y(t) - \widehat{\varphi}_k^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}_g(k)$. Thus, we can obtain the particle swarm optimization iterative (PSOI) identification algorithm as follows:

$$\widehat{\boldsymbol{\theta}}_i(k+1) = \widehat{\boldsymbol{\theta}}_i(k) + \widehat{\mathbf{Q}}_i(k-1), \quad i = 1, 2, \dots, M, \tag{24}$$

$$\begin{aligned}\widehat{\mathbf{Q}}_i(k+1) &= \beta\widehat{\mathbf{Q}}_i(k) + \xi_1\zeta_1\left(\widehat{\boldsymbol{\theta}}_{ih}(k) - \widehat{\boldsymbol{\theta}}_i(k)\right) \\ &\quad + \xi_2\zeta_2\left(\widehat{\boldsymbol{\theta}}_g(k) - \widehat{\boldsymbol{\theta}}_i(k)\right), \quad i = 1, 2, \dots, M,\end{aligned} \tag{25}$$

$$\begin{aligned}\widehat{\varphi}_k(t) = \big[&-\widehat{x}_{k-1}(t-1), \dots, -\widehat{x}_{k-1}(t-n_a), \\ &u(t-1), \dots, u(t-n_b), \widehat{x}_{k-1}^2(t), \\ &\widehat{x}_{k-1}^3(t), \dots, \widehat{x}_{k-1}^{nr}(t), \widehat{v}_{k-1}(t-1), \\ &\widehat{v}_{k-1}(t-2), \dots, \widehat{v}_{k-1}(t-n_d)\big]^{\mathrm{T}},\end{aligned} \tag{26}$$

$$\hat{\varphi}_{1,k}(t) = \left[-\hat{x}_{k-1}(t-1), \ldots, -\hat{x}_{k-1}(t-n_a), u(t-1), \ldots, u(t-n_b)\right]^{\mathrm{T}}, \tag{27}$$

$$\hat{x}_k(t) = \hat{\varphi}_{1,k}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_{g1}(k), \tag{28}$$

$$\hat{\boldsymbol{\theta}}_{ih}(k) = \left[\hat{a}_{1ih}(k), \ldots, \hat{a}_{n_a ih}(k), \hat{b}_{1ih}(k), \ldots, \hat{b}_{n_b ih}(k), \hat{\gamma}_{2ih}(k), \ldots, \hat{\gamma}_{n_\gamma ih}(k), \hat{d}_{1ih}(k), \ldots, \hat{b}_{n_d ih}(k)\right]^{\mathrm{T}}, \tag{29}$$

$$\hat{\boldsymbol{\theta}}_g(k) = \left[\hat{a}_{1g}(k), \cdots, \hat{a}_{n_a g}(k), \hat{b}_{1g}(k), \cdots, \hat{b}_{n_b g}(k), \hat{\gamma}_{2g}(k), \cdots, \hat{\gamma}_{n_\gamma g}(k), \hat{d}_{1g}(k), \cdots, \hat{d}_{n_d g}(k)\right]^{\mathrm{T}}, \tag{30}$$

$$\mathbf{Y}(p) = [y(p), y(p-1), \ldots, y(1)]^{\mathrm{T}}, \tag{31}$$

$$\hat{\boldsymbol{\Phi}}_k(p) = [\hat{\varphi}_k(p), \hat{\varphi}_k(p-1), \ldots, \hat{\varphi}_k(1)], \tag{32}$$

$$\hat{\boldsymbol{\theta}}_{ih}(k) = \arg\min \left(\left\|\mathbf{Y}(p) - \hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(p)\hat{\boldsymbol{\theta}}_i(k)\right\|, \left\|\mathbf{Y}(p) - \hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(p)\hat{\boldsymbol{\theta}}_{ih}(k-1)\right\|\right), \tag{33}$$

$$\hat{\boldsymbol{\theta}}_g(k) = \arg\min \left(\left\|\mathbf{Y}(p) - \hat{\boldsymbol{\Phi}}_k^{\mathrm{T}}(p)\hat{\boldsymbol{\theta}}_{ih}(k)\right\|\right), \tag{34}$$

$$\hat{\boldsymbol{\theta}}_{g1}(k) = \left[\hat{a}_{1g}(k), \ldots, \hat{a}_{n_a g}(k), \hat{b}_{1g}(k), \ldots, \hat{b}_{n_b g}(k)\right]^{\mathrm{T}}, \tag{35}$$

$$\hat{v}_k(t) = y(t) - \hat{\varphi}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_g(k). \tag{36}$$

The steps of the PSOI algorithm are listed as follows:

(1) Let $k = 0$, set the initial values as $\hat{\boldsymbol{\theta}}_i(0)$, $\hat{\mathbf{Q}}_i(0)$, $\hat{\boldsymbol{\theta}}_{ih}(0)$, and $\hat{\boldsymbol{\theta}}_g(0)$, $i = 1, 2, \ldots, M$. Set the initial factor β, ξ_1, ξ_2 and give a small positive number ε. Set $\hat{x}_0(t) = 1/p_0$, $p_0 = 10^6$, and $\hat{v}_0(t) = 0$.

(2) Collect the input and output data $u(t)$ and $y(t)$, $t = 1, 2, \ldots, p$, form $\hat{\varphi}_{1,k}(t)$ by (27) and $\hat{\varphi}_k(t)$ by (26). Construct $\mathbf{Y}(p)$ and $\boldsymbol{\Phi}_k(p)$ by (31) and (32), respectively.

(3) Update the velocity of each particle $\hat{\mathbf{Q}}_i(k+1)$, $i = 1, 2, \ldots, M$, according to (25).

(4) Update the position of each particle $\hat{\boldsymbol{\theta}}_i(k+1)$ by (24).

(5) Compute the best position of each particle $\hat{\boldsymbol{\theta}}_{ih}(k+1)$ by (33).

(6) Determine the best position of all the particles $\hat{\boldsymbol{\theta}}_g(k+1)$ by (34).

(7) Compute $\hat{v}_{k+1}(t)$ by (36). Form $\hat{\boldsymbol{\theta}}_{g1}(k+1)$ by (35), compute $\hat{x}_{k+1}(t)$ by (28).

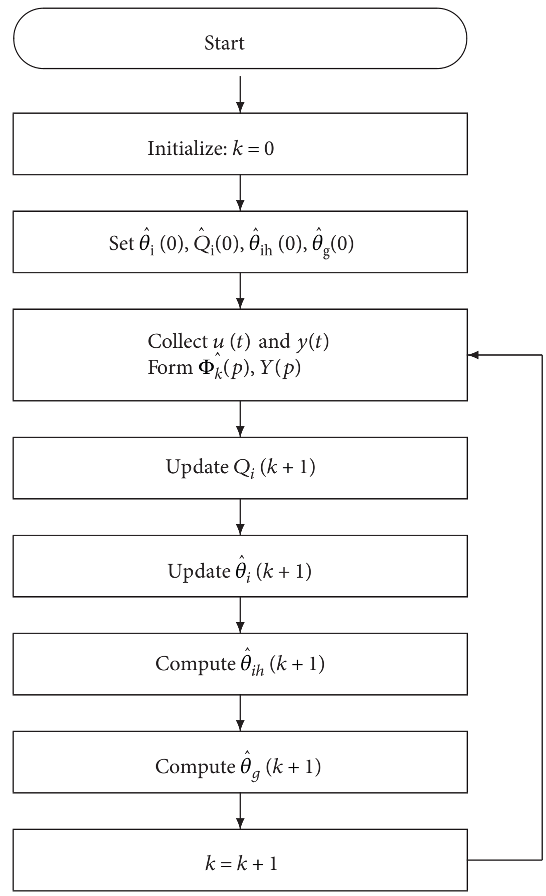

Figure 2: The flowchart of PSOI algorithm.

(8) Compare $\hat{\boldsymbol{\theta}}_g(k+1)$ and $\hat{\boldsymbol{\theta}}_g(k)$: if $\|\hat{\boldsymbol{\theta}}_g(k+1) - \hat{\boldsymbol{\theta}}_g(k)\| \leq \varepsilon$, then terminate the procedure and obtain the estimate $\hat{\boldsymbol{\theta}}_g(k+1)$; otherwise, increase k by 1 and go to Step 2.

The flowchart of PSOI algorithm is shown in Figure 2.

Remark 1. The major factors that influence the performance of the particle swarm optimization include ξ_1,ξ_2, and β. ξ_1 and ξ_2 are positive constants between 0 and 2. ξ_1 is the step size that adjusts the particle to its own best position. ξ_2 is the step size that regulates the particle to the global best position. β is called the inertia factor and is an important adjusting parameter of the PSOI algorithm. A larger β can facilitate global optimization; otherwise, a smaller one can facilitate local optimization. It can be chosen as a constant between 0.1 and 0.9 generally. ζ_1 and ζ_2 are two independent random numbers uniformly distributed in the range of [0, 1].

4. Gradient Iterative Algorithm

The gradient search is a very basic and ancient search method [37, 38]. It is widely used in parameter identification of nonlinear systems [39–41]. In the following, based on the gradient search principle, a gradient iterative identification algorithm for Wiener nonlinear model is derived.

Consider the latest p group data from $i = t - p + 1$ to $i = t$ and define the stacked output vector $\mathbf{Y}(t)$, the stacked information matrix $\mathbf{\Phi}(t)$, and the stacked noise vector $\mathbf{V}(t)$ as follows:

$$\begin{aligned}\mathbf{Y}(t) &:= [y(t), y(t-1), \dots, y(t-p+1)]^{\mathrm{T}} \in \mathbb{R}^{p},\\ \mathbf{\Phi}(t) &:= [\varphi(t), \varphi(t-1), \dots, \varphi(t-p+1)]^{\mathrm{T}} \in \mathbb{R}^{p\times n},\\ \mathbf{V}(t) &:= [v(t), v(t-1), \dots, v(t-p+1)]^{\mathrm{T}} \in \mathbb{R}^{p}.\end{aligned} \tag{37}$$

From (11), we have

$$\mathbf{Y}(t) = \mathbf{\Phi}(t)\boldsymbol{\theta} + V(t). \tag{38}$$

Define the criterion function

$$J_1(\boldsymbol{\theta}) := \|\mathbf{Y}(t) - \mathbf{\Phi}(t)\boldsymbol{\theta}\|^2. \tag{39}$$

Let $k = 1, 2, 3, \dots, n$ as an iterative variable and $\widehat{\boldsymbol{\theta}}_k(t)$ is the kth iterative estimation of parameter vector $\boldsymbol{\theta}$ at time t. For the optimization problem (39), the gradient iterative algorithm is obtained by using the negative gradient search

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_k(t) &= \widehat{\boldsymbol{\theta}}_{k-1}(t) - \frac{\mu_k(t)}{2}\ \mathrm{grad}\left[J_1\left(\widehat{\boldsymbol{\theta}}_{k-1}(t)\right)\right]\\ &= \widehat{\boldsymbol{\theta}}_{k-1}(t) + \mu_k(t)\Phi^{\mathrm{T}}(t)\left[\mathbf{Y}(t) - \mathbf{\Phi}(t)\widehat{\boldsymbol{\theta}}_{k-1}(t)\right],\end{aligned} \tag{40}$$

where $\mu_k(t)$ is the iterative step-size. However, in the upper formula of (40), the gradient iterative estimate $\widehat{\boldsymbol{\theta}}_k(t)$ impossible to calculate because the stacked information vector $\mathbf{\Phi}(t)$ contains unknown intermediate variables $x(t)$ and $v(t)$. The solution is to replace the unknown variables $x(t)$ and $v(t)$ by $\widehat{x}_{k-1}(t)$ and $\widehat{v}_{k-1}(t)$, respectively. Let $\widehat{\varphi}_{1,k}(t)$, $\widehat{\varphi}_{2,k}(t)$, and $\widehat{\varphi}_k(t)$ denote the estimates of $\varphi_1(t)$, $\varphi_2(t)$, and $\varphi(t)$ at iteration k, respectively

$$\begin{aligned}\widehat{\varphi}_{1,k}(t) &:= \left[-\widehat{x}_{k-1}(t-1), \dots, -\widehat{x}_{k-1}(t-n_a),\right.\\ &\qquad \left. u(t-1), \dots, u(t-n_b)\right]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_b},\\ \widehat{\varphi}_{2,k}(t) &:= \left[\widehat{x}^2_{k-1}(t), \widehat{x}^3_{k-1}(t), \dots, \widehat{x}^{n_r}_{k-1}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{n_\gamma-1},\\ \widehat{\varphi}_{d,k}(t) &:= [\widehat{v}_{k-1}(t-1), \widehat{v}_{k-1}(t-2), \dots, \widehat{v}_{k-1}(t-n_d)]^{\mathrm{T}} \in \mathbb{R}^{n_d},\\ \widehat{\varphi}_k(t) &:= \left[\widehat{\varphi}^{\mathrm{T}}_{1,k}(t), \widehat{\varphi}^{\mathrm{T}}_{2,k}(t), \widehat{\varphi}^{\mathrm{T}}_{d,k}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{n}.\end{aligned} \tag{41}$$

Let $\widehat{\mathbf{\Phi}}_k(t)$ denote the estimates of $\mathbf{\Phi}(t)$ at iteration k

$$\widehat{\mathbf{\Phi}}_k(t) := [\widehat{\varphi}_k(t), \widehat{\varphi}_k(t-1), \dots, \widehat{\varphi}_k(t-p+1)]^{\mathrm{T}} \in \mathbb{R}^{p\times n}, \tag{42}$$

and let $\widehat{\boldsymbol{\theta}}_{1,k}(t)$ denote the estimates of $\boldsymbol{\theta}_1$ at iteration k

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{1,k}(t) &:= \left[\widehat{a}_{1,k}(t), \widehat{a}_{2,k}(t), \dots, \widehat{a}_{n_a,k}(t), \widehat{b}_{1,k}(t),\right.\\ &\qquad \left.\widehat{b}_{2,k}(t), \dots, \widehat{b}_{n_b,k}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_b}.\end{aligned} \tag{43}$$

Thus, $\widehat{x}_k(t)$ can be calculated by the following:

$$\widehat{x}_k(t) := \widehat{\varphi}^{\mathrm{T}}_{1,k}(t)\widehat{\boldsymbol{\theta}}_{1,k}(t). \tag{44}$$

Define $\widehat{\boldsymbol{\theta}}_k(t) = [\boldsymbol{\theta}^{\mathrm{T}}_{1,k}(t), \boldsymbol{\theta}^{\mathrm{T}}_{2,k}(t), \boldsymbol{\theta}^{\mathrm{T}}_{d,k}(t)]^{\mathrm{T}}$ as the estimates of $\boldsymbol{\theta} = [\boldsymbol{\theta}^{\mathrm{T}}_1, \boldsymbol{\theta}^{\mathrm{T}}_2, \boldsymbol{\theta}^{\mathrm{T}}_d]^{\mathrm{T}}$ at iteration k and then the estimates of $v_k(t)$ can be obtained by the following:

$$\widehat{v}_k(t) := y(t) - \widehat{\varphi}^{\mathrm{T}}_k(t)\widehat{\boldsymbol{\theta}}_k(t). \tag{45}$$

Replacing $\mathbf{\Phi}(t)$ by $\widehat{\mathbf{\Phi}}_k(t)$ in (40) and rewriting $\widehat{\boldsymbol{\theta}}_k(t)$ give

$$\widehat{\boldsymbol{\theta}}_k(t) = \widehat{\boldsymbol{\theta}}_{k-1}(t) + \mu_k(t)\widehat{\mathbf{\Phi}}^{\mathrm{T}}_k(t)\left[\mathbf{Y}(t) - \widehat{\mathbf{\Phi}}_k(t)\widehat{\boldsymbol{\theta}}_{k-1}(t)\right]^{\mathrm{T}}. \tag{46}$$

Thus, we can obtain the gradient iterative (GI) estimation algorithm for Wiener models

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_k(t) &= \widehat{\boldsymbol{\theta}}_{k-1}(t) + \mu_k(t)\widehat{\mathbf{\Phi}}^{\mathrm{T}}_k(t)\\ &\quad \cdot \left[\mathbf{Y}(t) - \widehat{\mathbf{\Phi}}_k(t)\widehat{\boldsymbol{\theta}}_{k-1}(t)\right], \quad k = 1, 2, 3, \dots,\end{aligned} \tag{47}$$

$$\widehat{\mathbf{\Phi}}_k(t) = [\widehat{\varphi}_k(t), \widehat{\varphi}_k(t-1), \dots, \widehat{\varphi}_k(t-p+1)]^{\mathrm{T}}, \tag{48}$$

$$\mathbf{Y}(t) = [y(t), y(t-1), \dots, y(t-p+1)]^{\mathrm{T}}, \tag{49}$$

$$\widehat{\varphi}_k(t) = \left[\widehat{\varphi}^{\mathrm{T}}_{1,k}(t), \widehat{\varphi}^{\mathrm{T}}_{2,k}(t), \widehat{\varphi}^{\mathrm{T}}_{d,k}(t)\right]^{\mathrm{T}}, \tag{50}$$

$$\begin{aligned}\widehat{\varphi}_{1,k}(t) &= \left[-\widehat{x}_{k-1}(t-1), \dots, -\widehat{x}_{k-1}(t-n_a),\right.\\ &\qquad \left.u(t-1), \dots, u(t-n_b)\right]^{\mathrm{T}},\end{aligned} \tag{51}$$

$$\widehat{\varphi}_{2,k}(t) = \left[\widehat{x}^2_{k-1}(t), \widehat{x}^3_{k-1}(t), \dots, \widehat{x}^{n_r}_{k-1}(t)\right]^{\mathrm{T}}, \tag{52}$$

$$\widehat{\varphi}_{d,k}(t) = [\widehat{v}_{k-1}(t-1), \widehat{v}_{k-1}(t-2), \cdots, \widehat{v}_{k-1}(t-n_d)]^{\mathrm{T}}, \tag{53}$$

$$\widehat{v}_k(t) = y(t) - \widehat{\varphi}^{\mathrm{T}}_k(t)\widehat{\boldsymbol{\theta}}_k(t), \tag{54}$$

$$\widehat{x}_k(t) = \widehat{\varphi}^{\mathrm{T}}_{1,k}(t)\widehat{\boldsymbol{\theta}}_{1,k}(t), \tag{55}$$

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{1,k}(t) &= \left[\widehat{a}_{1,k}(t), \widehat{a}_{2,k}(t), \dots, \widehat{a}_{n_a,k}(t),\right.\\ &\qquad \left.\widehat{b}_{1,k}(t), \widehat{b}_{2,k}(t), \dots, \widehat{b}_{n_b,k}(t)\right]^{\mathrm{T}},\end{aligned} \tag{56}$$

$$0 < \mu_k(t) \le \frac{2}{\lambda_{\max}\left[\widehat{\mathbf{\Phi}}^{\mathrm{T}}_k(t)\widehat{\mathbf{\Phi}}_k(t)\right]}. \tag{57}$$

The steps of the GI algorithm are listed as follows:

(1) Let $k = 1$, $\widehat{\boldsymbol{\theta}}_0(t) = 1/p_0$, $\widehat{x}_0(t) = 1/p_0$, and $p_0 = 10^6$ and give a small positive number ε.

(2) Collect the input and output data $\mu(t)$ and $y(t)$ and form $\mathbf{Y}(t)$ by (49) and $\widehat{\varphi}_{1,k}(t)$ by (51).

(3) Form $\widehat{\boldsymbol{\theta}}_{1,k}(t)$ by (56), compute $\widehat{x}_k(t)$ by (55), and form $\widehat{\varphi}_{2,k}(t)$ by (52).

(4) Compute $\widehat{v}_k(t)$ by (54) and form $\widehat{\varphi}_{d,k}(t)$ by (53).

(5) Form $\widehat{\varphi}_k(t)$ by (50) and form $\widehat{\mathbf{\Phi}}_k(t)$ by (48).

(6) Choose a suitable $\mu_k(t)$ by (57) and compute $\widehat{\boldsymbol{\theta}}_k(t)$ by (47).

Table 1: The GI estimates and errors ($\sigma^2 = 0.10^2$).

k	α_1	α_2	b_1	b_2	γ_2	γ_3	d_1	d_2	δ (%)
1	0.00000	0.00000	0.58390	0.68393	0.00000	0.00000	−0.04169	−0.02620	86.40246
5	−0.02316	0.00425	0.61143	0.71531	0.02417	0.09941	0.13680	−0.03462	80.62844
10	−0.05236	0.01690	0.65537	0.75735	0.06610	0.28009	0.31492	−0.05635	72.44222
50	−0.19707	0.17193	0.77823	0.85545	0.55958	1.11096	0.34000	−0.01365	31.03963
100	−0.26799	0.23558	0.76378	0.81452	0.78248	1.08091	0.26414	−0.03625	18.93492
500	−0.40949	0.33611	0.75852	0.64487	0.96516	1.14748	0.20235	−0.07345	2.45484
700	−0.42263	0.34486	0.75695	0.62722	0.98006	1.16077	0.20205	−0.07432	1.59965
True values	−0.43000	0.35000	0.76000	0.62000	0.98000	1.15000	0.20000	−0.10000	0.00000

Table 2: The PSOI estimates and errors ($M = 60$).

k	α_1	α_2	b_1	b_2	γ_2	γ_3	d_1	d_2	δ (%)
10	−0.70817	0.59244	0.68145	0.44089	0.69816	1.06690	0.32058	−0.04582	27.79105
20	−0.74261	0.64034	0.84768	0.63585	0.98413	1.08805	0.19259	0.03973	24.33881
30	−0.58252	0.36827	0.80718	0.54617	0.99907	1.13505	0.10336	−0.10655	10.69988
60	−0.42770	0.35023	0.74459	0.60637	0.98132	1.15542	0.19161	−0.09934	1.21353
90	−0.43027	0.35147	0.75416	0.61506	0.97971	1.15157	0.19172	−0.09235	0.72682
120	−0.43132	0.35162	0.75594	0.61617	0.97964	1.15089	0.19158	−0.09109	0.71978
True values	−0.43000	0.35000	0.76000	0.62000	0.98000	1.15000	0.20000	−0.10000	0.00000

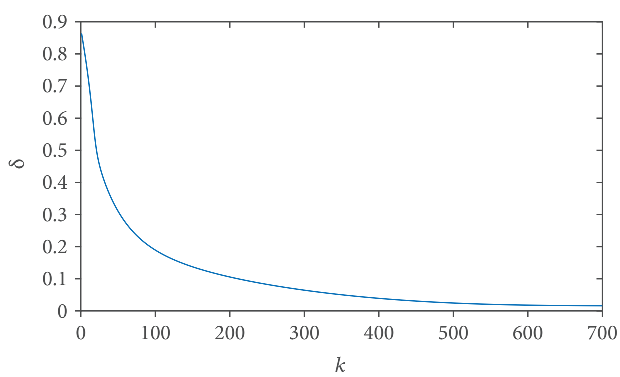

Figure 3: The GI estimation errors versus k.

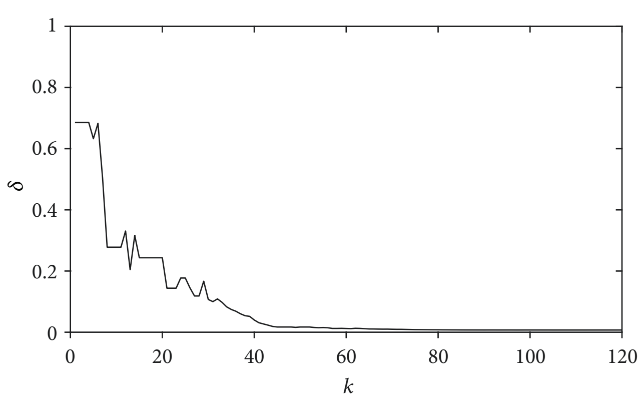

Figure 4: The PSOI estimation errors versus k.

(7) Compare $\widehat{\boldsymbol{\theta}}_k(t)$ and $\widehat{\boldsymbol{\theta}}_{k-1}(t)$: if $\|\widehat{\boldsymbol{\theta}}_k(t) - \widehat{\boldsymbol{\theta}}_{k-1}(t)\| \le \varepsilon$, then terminate the procedure and obtain $\widehat{\boldsymbol{\theta}}_k(t)$; otherwise, increase k by 1 and go to Step 2.

5. Examples

Consider the following Wiener nonlinear systems:

$$
\begin{aligned}
y(t) &= f(x(t)) + D(z)v(t),\\
x(t) &= \frac{B(z)}{A(z)}u(t),\\
A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} = 1 - 0.43z^{-1} + 0.35z^{-2},\\
B(z) &= b_1 z^{-1} + b_2 z^{-2} = 0.76z^{-1} + 0.62z^{-1},\\
D(z) &= 1 + d_1 z^{-1} + d_2 z^{-2} = 1 + 0.20z^{-1} - 0.10z^{-2},\\
f(x(t)) &= \gamma_1 x(t) + \gamma_2 x^2(t) + \gamma_3 x^3(t)\\
&= x(t) + 0.98x^2(t) + 1.15x^3(t),\\
\boldsymbol{\theta} &= [a_1, a_2, b_1, b_2, \gamma_2, \gamma_3, d_1, d_2]^{\mathrm{T}}\\
&= [-0.43, 0.35, 0.76, 0.62, 0.98, 1.15, 0.20, -0.10]^{\mathrm{T}}.
\end{aligned}
\tag{58}
$$

In simulation, the input $u(t)$ is taken as an uncorrelated stochastic signal sequence with zero mean and unit variance and $v(t)$ as a Gaussian white noise sequence with zero mean and variance $\sigma^2 = 0.10^2$. Applying the GI algorithm and the PSOI algorithm to estimate the parameters of this system, the parameter estimates and their errors are shown in Tables 1 and 2 and Figures 3 and 4. In the PSOI algorithm simulation, each swarm generation contains 10 position particles. The coefficients β, ξ_1, and ξ_2 in (25) are set to 0.8, 1.2, and 1.8, respectively.

From the simulation results in Tables 1 and 2 and Figures 3 and 4, we can draw the following conclusions:

(i) As k increases, the parameter estimation errors given by the GI algorithm and PSOI algorithm gradually become smaller (see Tables 1 and 2).

(ii) The PSOI algorithm has a faster convergence rate than the GI algorithm (see Figures 3 and 4).

(iii) The PSOI algorithm has a higher estimation accuracy than the GI algorithm, which can be seen from Tables 1 and 2.

6. Conclusions

In this paper, we derived the particle swarm optimization iterative algorithm and the gradient iterative algorithm for Wiener nonlinear systems. Compared with the gradient iterative algorithm, the particle swarm optimization algorithm has a higher estimation accuracy and has a faster convergence rate. The proposed approaches in the paper can be combined with other mathematical tools [42–47] to study the performances of some parameter estimation algorithms and can be applied to other multivariable systems with different structures and disturbance noises and other literature [48–52].

Data Availability

The relevant data can be obtained by email to the author.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61403217) and the Jiangsu Province Postdoctoral Research Funding Plan (1601129B).

References

[1] J. Na, J. Yang, X. Wu, and Y. Guo, "Robust adaptive parameter estimation of sinusoidal signals," *Automatica*, vol. 53, pp. 376–384, 2015.

[2] D. Q. Wang, Z. Zhang, and J. Y. Yuan, "Maximum likelihood estimation method for dual-rate Hammerstein systems," *International Journal of Control Automation and Systems*, vol. 15, no. 2, pp. 698–705, 2017.

[3] J. Na, A. S. Chen, G. Herrmann, R. Burke, and C. Brace, "Vehicle engine torque estimation via unknown input observer and adaptive parameter estimation," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 409–422, 2018.

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

[5] L. Xu, F. Ding, Y. Gu, A. Alsaedi, and T. Hayat, "A multi-innovation state and parameter estimation algorithm for a state space system with d-step state-delay," *Signal Processing*, vol. 140, pp. 97–103, 2017.

[6] M. Li and X. Liu, "The least squares based iterative algorithms for parameter estimation of a bilinear system with autoregressive noise using the data filtering technique," *Signal Processing*, vol. 147, pp. 23–34, 2018.

[7] X. Zhang, F. Ding, F. E. Alsaadi, and T. Hayat, "Recursive parameter identification of the dynamical models for bilinear state space systems," *Nonlinear Dynamics*, vol. 89, no. 4, pp. 2415–2429, 2017.

[8] X. Zhang, L. Xu, F. Ding, and T. Hayat, "Combined state and parameter estimation for a bilinear state space system with moving average noise," *Journal of the Franklin Institute*, vol. 355, no. 6, pp. 3079–3103, 2018.

[9] D. Wang, L. Li, Y. Ji, and Y. Yan, "Model recovery for Hammerstein systems using the auxiliary model based orthogonal matching pursuit method," *Applied Mathematical Modelling*, vol. 54, pp. 537–550, 2018.

[10] J. Li, W. X. Zheng, J. Gu, and L. Hua, "A recursive identification algorithm for wiener nonlinear systems with linear state-space subsystem," *Circuits, Systems and Signal Processing*, vol. 37, no. 6, pp. 2374–2393, 2018.

[11] M. Gan, C. L. P. Chen, G. Y. Chen, and L. Chen, "On some separated algorithms for separable nonlinear least squares problems," *IEEE Transactions on Cybernetics*, pp. 1–9, 2018.

[12] A. Radouane, F. Giri, F. Ikhouane, A. A. Tarek, F. Z. Chaoui, and B. Adil, "System identification of a class of wiener systems with hysteretic nonlinearities," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 3, pp. 332–359, 2017.

[13] G. Bottegal, R. Castro-Garcia, and J. A. K. Suykens, "A two-experiment approach to wiener system identification," *Automatica*, vol. 93, pp. 282–289, 2018.

[14] J. Kou, W. Zhang, and M. Yin, "Novel wiener models with a time-delayed nonlinear block and their identification," *Nonlinear Dynamics*, vol. 85, no. 4, pp. 2389–2404, 2016.

[15] Z. Wang and X. Luo, "A novel weight function-based robust iterative learning identification method for discrete Box-Jenkins models with Student's t-distribution noises," *Journal of the Franklin Institute*, vol. 354, no. 18, pp. 8645–8658, 2017.

[16] J. Lardiès, "Modal parameter identification by an iterative approach and by the state space model," *Mechanical Systems and Signal Processing*, vol. 95, pp. 239–251, 2017.

[17] J. Li, W. X. Zheng, J. Gu, and L. Hua, "Parameter estimation algorithms for Hammerstein output error systems using Levenberg-Marquardt optimization method with varying interval measurements," *Journal of the Franklin Institute*, vol. 354, no. 1, pp. 316–331, 2017.

[18] F. Ding, F. Wang, L. Xu, and M. Wu, "Decomposition based least squares iterative identification algorithm for multivariate pseudo-linear ARMA systems using the data filtering," *Journal of the Franklin Institute*, vol. 354, no. 3, pp. 1321–1339, 2017.

[19] F. Ding, D. Meng, J. Dai, Q. Li, A. Alsaedi, and T. Hayat, "Least squares based iterative parameter estimation algorithm for stochastic dynamical systems with ARMA noise using the model

equivalence," *International Journal of Control, Automation and Systems*, vol. 16, no. 2, pp. 630–639, 2018.

[20] F. Ding, H. Chen, L. Xu, J. Dai, Q. Li, and T. Hayat, "A hierarchical least squares identification algorithm for Hammerstein nonlinear systems using the key term separation," *Journal of the Franklin Institute*, vol. 355, no. 8, pp. 3737–3752, 2018.

[21] 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 and Applications*, vol. 12, no. 7, pp. 892–899, 2018.

[22] J. Ding, "Recursive and iterative least squares parameter estimation algorithms for multiple-input-output-error systems with autoregressive noise," *Circuits, Systems and Signal Processing*, vol. 37, no. 5, pp. 1884–1906, 2018.

[23] S. Sadeghi, H. Parvin, and F. Rad, "Particle swarm optimization algorithm for dynamic environments," *Lecture Notes in Computer Science*, vol. 9413, pp. 260–269, 2015.

[24] Y. X. Zheng and Y. Liao, "Parameter identification of nonlinear dynamic systems using an improved particle swarm optimization," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 19, pp. 7865–7874, 2016.

[25] G. Xu and G. Yu, "On convergence analysis of particle swarm optimization algorithm," *Journal of Computational and Applied Mathematics*, vol. 333, no. 1, pp. 65–73, 2018.

[26] Y. Wang, W. G. Zhang, F. Li, D. G. Huang, and Y. Li, "Nash equilibrium strategies approach for aerial combat based on elite re-election particle swarm optimization," *Control Theory and Applications*, vol. 32, no. 7, pp. 857–865, 2015.

[27] G.-H. Lin, J. Zhang, and Z.-H. Liu, "Hybrid particle swarm optimization with differential evolution for numerical and engineering optimization," *International Journal of Automation and Computing*, vol. 15, no. 1, pp. 103–114, 2018.

[28] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi, "Fuzzy self-tuning PSO: a settings-free algorithm for global optimization," *Swarm and Evolutionary Computation*, vol. 39, pp. 70–85, 2018.

[29] J. Liu, H. Zhang, K. He, and S. Jiang, "Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem," *Expert Systems with Applications*, vol. 102, pp. 179–192, 2018.

[30] H. Modares, A. Alfi, and M. M. Fateh, "Parameter identification of chaotic dynamic systems through an improved particle swarm optimization," *Expert Systems with Applications*, vol. 37, no. 5, pp. 3714–3720, 2010.

[31] J. Chen and X. Wang, "Identification of Hammerstein systems with continuous nonlinearity," *Information Processing Letters*, vol. 115, no. 11, pp. 822–827, 2015.

[32] A. S. Wu and R. K. Lindsay, "A comparison of the fixed and floating building block representation in the genetic algorithm," *Evolutionary Computation*, vol. 4, no. 2, pp. 169–193, 1996.

[33] S. R. Balseiro, I. Loiseau, and J. Ramonet, "An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows," *Computers & Operations Research*, vol. 38, no. 6, pp. 954–966, 2011.

[34] A. Alfi and H. Modares, "System identification and control using adaptive particle swarm optimization," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1210–1221, 2011.

[35] M. Pulido and P. Melin, "Optimization of ensemble neural networks with fuzzy integration using the particle swarm algorithm for the US dollar/MX time series prediction," *Intelligent Automation & Soft Computing*, vol. 20, no. 1, pp. 1–7, 2015.

[36] A. Alfi and M. M. Fateh, "Intelligent identification and control using improved fuzzy particle swarm optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12312–12317, 2011.

[37] M. Chen, F. Ding, L. Xu, T. Hayat, and A. Alsaedi, "Iterative identification algorithms for bilinear-in-parameter systems with autoregressive moving average noise," *Journal of the Franklin Institute*, vol. 354, no. 17, pp. 7885–7898, 2017.

[38] L. Xu, "The parameter estimation algorithms based on the dynamical response measurement data," *Advances in Mechanical Engineering*, vol. 9, no. 11, 2017.

[39] M. Li, X. Liu, and F. Ding, "Least-squares-based iterative and gradient-based iterative estimation algorithms for bilinear systems," *Nonlinear Dynamics*, vol. 89, no. 1, pp. 197–211, 2017.

[40] M. Li, X. Liu, and F. Ding, "The maximum likelihood least squares based iterative estimation algorithm for bilinear systems with autoregressive moving average noise," *Journal of the Franklin Institute*, vol. 354, no. 12, pp. 4861–4881, 2017.

[41] M. Li, X. Liu, and F. Ding, "The gradient based iterative estimation algorithms for bilinear systems with autoregressive noise," *Circuits, Systems and Signal Processing*, vol. 36, no. 11, pp. 4541–4568, 2017.

[42] F. Liu, Q. Xue, and K. Yabuta, "Rough maximal singular integral and maximal operators supported by subvarieties on Triebel-Lizorkin spaces," *Nonlinear Analysis*, vol. 171, pp. 41–72, 2018.

[43] F. Liu, "Continuity and approximate differentiability of multisublinear fractional maximal functions," *Mathematical Inequalities & Applications*, vol. 21, no. 1, pp. 25–40, 2018.

[44] F. Liu and H. Wu, "Singular integrals related to homogeneous mappings in Triebel-Lizorkin spaces," *Journal of Mathematical Inequalities*, vol. 11, no. 4, pp. 1075–1097, 2017.

[45] F. Liu and H. X. Wu, "Regularity of discrete multisublinear fractional maximal functions," *Science China–Mathematics*, vol. 60, no. 8, pp. 1461–1476, 2017.

[46] F. Liu and H. Wu, "On the regularity of maximal operators supported by submanifolds," *Journal of Mathematical Analysis and Applications*, vol. 453, no. 1, pp. 144–158, 2017.

[47] F. Liu, "On the Triebel-Lizorkin space boundedness of Marcinkiewicz integrals along compound surfaces," *Mathematical Inequalities & Applications*, vol. 20, no. 2, pp. 515–535, 2017.

[48] K. Tiels and J. Schoukens, "Wiener system identification with generalized orthonormal basis functions," *Automatica*, vol. 50, no. 12, pp. 3147–3154, 2014.

[49] Q. Chen, L. Shi, J. Na, X. Ren, and Y. Nan, "Adaptive echo state network control for a class of pure-feedback systems with input and output constraints," *Neurocomputing*, vol. 275, no. 31, pp. 1370–1382, 2018.

[50] N. Zhao, Y. Chen, R. Liu, M. Wu, and W. Xiong, "Monitoring strategy for relay incentive mechanism in cooperative communication networks," *Computers & Electrical Engineering*, vol. 60, pp. 14–29, 2017.

[51] P. Gong, W.-Q. Wang, F. Li, and H. C. So, "Sparsity-aware transmit beamspace design for FDA-MIMO radar," *Signal Processing*, vol. 144, pp. 99–103, 2018.

[52] Z. Rao, C. Zeng, M. Wu, Z. Wang, N. Zhan, and M. L. X. Wan, "Research on a handwritten character recognition algorithm based on an extended nonlinear kernel residual network," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 1, pp. 413–435, 2018.

Advances in
Operations Research
Advances in
Decision Sciences
Journal of
Applied Mathematics
The Scientific
World Journal
Journal of
Probability and Statistics
International
Journal of
Mathematics and
Mathematical
Sciences
Hindawi
Submit your manuscripts at
www.hindawi.com
Journal of
Optimization
International Journal of
Engineering
Mathematics
International Journal of
Analysis
Journal of
Complex Analysis
Advances in
Numerical Analysis
Mathematical Problems
in Engineering
International Journal of
Differential Equations
Discrete Dynamics in
Nature and Society
International Journal of
Stochastic Analysis
Journal of
Mathematics
Journal of
Function Spaces
Abstract and
Applied Analysis
Advances in
Mathematical Physics