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

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

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.
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:

\[
y(t) = f(x(t)) + D(z)v(t),
\]

\[
x(t) = \frac{B(z)}{A(z)} u(t),
\]

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

\[
A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n},
\]

\[
B(z) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n},
\]

\[
D(z) = 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_n z^{-n}.
\]

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

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

\[
x(t) = \frac{B(z)}{A(z)} u(t),
\]

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

\[
w(t) = D(z)v(t).
\]

The static nonlinear block is a nonlinear function

\[
f(x(t)) = y_1 f_1(x(t)) + y_2 f_2(x(t)) + \cdots + y_n f_n(x(t)),
\]

where the basis \(g = (f_1, f_2, \ldots, f_n)\) are known nonlinear functions of \(x(t),\) the unknown parameters \(y_i\) are the coefficients of the nonlinear functions and assume that the degree \(n_y\) is known. Without loss of generality, let the first coefficient of nonlinear block \(y_1\) be unity and rewrite the \(f(x(t))\) as

\[
f(x(t)) = x(t) + y_2 x^2(t) + \cdots + y_n x^n(t).
\]

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 \(\sigma^2.\) From (3), we have

\[
x(t) = [1 - A(z)]x(t) + B(z)u(t)
\]

\[
= -a_1 x(t - 1) - \cdots - a_n x(t - n)
\]

\[
+ b_1 u(t - 1) + \cdots + b_n u(t - n) = \phi^T_1(t) \theta_1,
\]

where

\[
\phi_1(t) = [-x(t - 1), \ldots, -x(t - n), u(t - 1), \ldots, u(t - n)]^T \in \mathbb{R}^{n \times n},
\]

\[
\theta_1 = [a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n]^T \in \mathbb{R}^{n \times n}.
\]

From (4), we can obtain

\[
w(t) = D(z)v(t)
\]

\[
= d_1 v(t - 1) + d_2 v(t - 2) + \cdots + d_n v(t - n) = \phi^T_d(t) \theta_d,
\]

where

\[
\phi_d(t) = [v(t - 1), v(t - 2), \ldots, v(t - n)]^T \in \mathbb{R}^{n \times n},
\]

\[
\theta_d = [d_1, d_2, \ldots, d_n]^T \in \mathbb{R}^{n \times n}.
\]

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

\[
y(t) = f(x(t)) + w(t)
\]

\[
= x(t) + y_2 x^2(t) + \cdots + y_n x^n(t) + D(z)v(t)
\]

\[
= \phi^T_1(t) \theta_1 + \phi^T_2(t) \theta_2 + \phi^T_d(t) \theta_d + v(t)
\]

\[
= \phi^T(t) \theta + v(t),
\]

where

\[
\phi_2(t) = [x^2(t), x^3(t), \ldots, x^n(t)]^T \in \mathbb{R}^{n \times n - 1},
\]

\[
\theta_2 = [y_2, y_3, \ldots, y_n]^T \in \mathbb{R}^{n \times n - 1},
\]

\[
\phi(t) = [\phi^T_1(t), \phi^T_2(t), \phi^T_d(t)]^T \in \mathbb{R}^n,
\]

\[
\theta = [\theta^T_1, \theta^T_2, \theta^T_d]^T \in \mathbb{R}^n,
\]

\[
n = n_a + n_b + n_y - 1 + n_d.
\]

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
\[
\Phi(t) = [-x(t-1), \ldots, -x(t-n_u), u(t-1), \ldots, \\
u(t-n_b), x^2(t), x^3(t), \ldots, x^n(t), \\
v(t-1), v(t-2), \ldots, v(t-n_d)]^T \in \mathbb{R}^n.
\]
(13)

Let \( p \) represent the data length. Define the stacked output
vector \( Y(p) \) and the stacked information matrix \( \Phi(p) \) as
\[
Y(p) = [v(p), v(p-1), \ldots, v(1)]^T \in \mathbb{R}^p,
\]
\[
\Phi(p) = [\Phi(p), \Phi(p-1), \ldots, \Phi(1)] \in \mathbb{R}^{np}.
\]
(14)

Define the independent position of each particle \( \hat{\theta}_i \) and
the independent velocity \( Q_i \) as follows:
\[
\hat{\theta}_i = [\hat{a}_{1,i}, \hat{a}_{2,i}, \ldots, \hat{a}_{n_i}, \hat{b}_{1,i}, \hat{b}_{2,i}, \ldots, \hat{b}_{n_i}, \hat{\gamma}_{2i}, \\
\hat{\gamma}_{3i}, \ldots, \hat{\gamma}_{n_i}, d_{1i}, d_{2i}, \ldots, \hat{d}_{n_i}]^T \in \mathbb{R}^n,
\]
\[
Q_i = [q_{1i}, q_{2i}, \ldots, q_{ni}]^T \in \mathbb{R}^n, \quad i = 1, 2, \ldots, M.
\]
(15)

Let \( \hat{\theta}(k) \) denote the estimates of \( \hat{\theta} \) at iteration \( k = 1, 2, 3, \ldots \). Define \( \hat{\theta}_{ih}(k) \) as the best position of each particle at
iteration \( k \)
\[
\hat{\theta}_{ih}(k) = [\hat{a}_{1,ih}(k), \ldots, \hat{a}_{n,ih}(k), \hat{b}_{1,ih}(k), \ldots, \hat{b}_{n,ih}(k), \\
\hat{\gamma}_{2,ih}(k), \ldots, \hat{\gamma}_{n,ih}(k), \hat{d}_{1,ih}(k), \ldots, \hat{d}_{n,ih}(k)]^T \in \mathbb{R}^n.
\]
(16)

Let \( \hat{\theta}_{1k}(t) \) and \( \hat{\theta}_k(t) \) denote the estimates of \( \hat{\theta}_1(t) \) and
\( \hat{\theta}(t) \) at iteration \( k \)
\[
\hat{\theta}_{1k}(t) = [-\hat{x}_{k-1}(t-1), \ldots, -\hat{x}_{k-1}(t-n_u), \\
u(t-1), \ldots, u(t-n_b)]^T \in \mathbb{R}^{n+nu},
\]
\[
\hat{\theta}_k(t) = [-\hat{x}_{k-1}(t-1), \ldots, -\hat{x}_{k-1}(t-n_u), \\
u(t-1), \ldots, u(t-n_b), \hat{x}_{k-1}^2(t), \\
\hat{x}_{k-1}^3(t), \ldots, \hat{x}_{k-1}^{n_u}(t), \hat{v}_{k-1}(t-1), \\
\hat{v}_{k-1}(t-2), \ldots, \hat{v}_{k-1}(t-n_d)]^T \in \mathbb{R}^n.
\]
(17)

Then, the estimates \( \hat{\Phi}_k(p) \) can be obtained as follows:
\[
\hat{\Phi}_k(p) = [\hat{\Phi}_k(p), \hat{\Phi}_k(p-1), \ldots, \hat{\Phi}_k(1)].
\]
(18)

According to the basic principle of the particle swarm
optimization algorithm, the best position of each particle \( \hat{\theta}_{ih}(k) \) satisfies
the following cost function:
\[
\hat{\theta}_{ih}(k) = \arg \min \left( \| Y(p) - \hat{\Phi}_k(p) \| \right).
\]
(19)

Let \( \hat{\theta}_g(k) \) denote the global best position of all the
particles
\[
\hat{\theta}_g(k) = \left[ \hat{a}_{1g}(k), \ldots, \hat{a}_{n,g}(k), \hat{b}_{1g}(k), \ldots, \hat{b}_{n,g}(k), \\
\hat{\gamma}_{2g}(k), \ldots, \hat{\gamma}_{n,g}(k), \hat{d}_{1g}(k), \ldots, \hat{d}_{n,g}(k) \right]^T \in \mathbb{R}^n,
\]
(20)

where \( \hat{\theta}_g(k) \) satisfies
\[
\hat{\theta}_g(k) = \arg \min \left( \| Y(p) - \hat{\Phi}_k(p) \| \right).
\]
(21)

Define \( \hat{\theta}_{g1}(k) = [\hat{a}_{1g}(k), \ldots, \hat{a}_{n,g}(k), \hat{b}_{1g}(k), \ldots, \hat{b}_{n,g}(k)]^T \).
According to (7), we can obtain the estimation of \( \hat{\xi}_k(t) \)
\[
\hat{\xi}_k(t) = \hat{\Phi}_k(t)^T \hat{\theta}_{g1}(k).
\]
(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:
\[
\hat{\theta}_i(k+1) = \hat{\theta}_i(k) + \hat{\xi}_i(k),
\]
\[
\hat{\xi}_i(k+1) = \beta \hat{\xi}_i(k) + \xi_1 \hat{\xi}_i \left( \hat{\theta}_i(k) - \hat{\theta}_i(k) \right)
\]
(23)

\[
+ \xi_2 \hat{\xi}_2 \left( \hat{\theta}_g(k) - \hat{\theta}_i(k) \right).
\]

Replacing \( \varphi(t) \) and \( \theta \) with \( \hat{\varphi}(t) \) and \( \hat{\theta}(k) \) in (11), we can obtain the estimate \( \hat{v}_k(t) = \varphi(t) - \hat{\Phi}_k(t)^T \hat{\theta}_g(k) \). Thus, we can obtain the particle swarm optimization iterative (PSOI) identity
algorithm as follows:
\[
\hat{\theta}_i(k + 1) = \hat{\theta}_i(k) + \hat{\xi}_i(k), \quad i = 1, 2, \ldots, M,
\]
(24)

\[
\hat{\xi}_i(k + 1) = \beta \hat{\xi}_i(k) + \xi_1 \hat{\xi}_i \left( \hat{\theta}_i(k) - \hat{\theta}_i(k) \right)
\]
(25)

\[
+ \xi_2 \hat{\xi}_2 \left( \hat{\theta}_g(k) - \hat{\theta}_i(k) \right), \quad i = 1, 2, \ldots, M,
\]

\[
\hat{\phi}_k(t) = [-\hat{x}_{k-1}(t-1), \ldots, -\hat{x}_{k-1}(t-n_u), \\
u(t-1), \ldots, u(t-n_b), \hat{x}_{k-1}^2(t), \\
\hat{x}_{k-1}^3(t), \ldots, \hat{x}_{k-1}^{n_u}(t), \hat{v}_{k-1}(t-1), \\
\hat{v}_{k-1}(t-2), \ldots, \hat{v}_{k-1}(t-n_d)]^T \in \mathbb{R}^n.
\]
(26)
\( \tilde{\varphi}_{1,k}(t) = \left[ -\tilde{x}_{k-1}(t-1), \ldots, -\tilde{x}_{k-1}(t-n_u), \right. \\
\left. u(t-1), \ldots, u(t-n_b) \right]^T, \) \hspace{1cm} (27)

\( \tilde{x}_k(t) = \tilde{\varphi}_{1,k}(t) \tilde{\theta}_{gl}(k), \) \hspace{1cm} (28)

\( \tilde{\theta}_{ih}(k) = \left[ \tilde{a}_{i1,h}(k), \ldots, \tilde{a}_{n_u,h}(k), \tilde{b}_{i1,h}(k), \ldots, \tilde{b}_{n_u,h}(k), \right. \\
\left. \tilde{\gamma}_{2i,h}(k), \ldots, \tilde{\gamma}_{n_u,h}(k), \tilde{d}_{i1,h}(k), \ldots, \tilde{d}_{n_u,h}(k) \right]^T, \) \hspace{1cm} (29)

\( \tilde{\theta}_{g}(k) = \left[ \tilde{a}_{i1,g}(k), \ldots, \tilde{a}_{n_u,g}(k), \tilde{b}_{i1,g}(k), \ldots, \tilde{b}_{n_u,g}(k), \right. \\
\left. \tilde{\gamma}_{2i,g}(k), \ldots, \tilde{\gamma}_{n_u,g}(k), \tilde{d}_{i1,g}(k), \ldots, \tilde{d}_{n_u,g}(k) \right]^T, \) \hspace{1cm} (30)

\( Y(p) = [y(p), y(p-1), \ldots, y(1)]^T, \) \hspace{1cm} (31)

\( \tilde{\Phi}_k(p) = [\tilde{\varphi}_k(p), \tilde{\varphi}_k(p-1), \ldots, \tilde{\varphi}_k(1)], \) \hspace{1cm} (32)

\( \tilde{\theta}_{ih}(k) = \text{arg min} \left[ \left\| Y(p) - \tilde{\Phi}_k(p)^T \tilde{\theta}_h(k) \right\|, \left\| Y(p) - \tilde{\Phi}_k(p)^T \tilde{\theta}_h(k-1) \right\| \right], \) \hspace{1cm} (33)

\( \tilde{\theta}_g(k) = \text{arg min} \left[ \left\| Y(p) - \tilde{\Phi}_k(p)^T \tilde{\theta}_g(k) \right\|, \right], \) \hspace{1cm} (34)

\( \tilde{\theta}_{gl}(k) = \left[ \tilde{a}_{i1,g}(k), \ldots, \tilde{a}_{n_u,g}(k), \tilde{b}_{i1,g}(k), \ldots, \tilde{b}_{n_u,g}(k) \right]^T, \) \hspace{1cm} (35)

\( \tilde{\varphi}_k(t) = y(t) - \tilde{\Phi}_k(t)^T \tilde{\theta}_g(k). \) \hspace{1cm} (36)

The steps of the PSOI algorithm are listed as follows:

1. Let \( k = 0, \) set the initial values as \( \tilde{\theta}_h(0), \tilde{\Phi}_k(0), \tilde{\theta}_{ih}(0), \) and \( \tilde{\theta}_g(0), i = 1, 2, \ldots, M. \) Set the initial factor \( \beta, \xi_1, \) and give a small positive number \( \epsilon. \) Set \( \tilde{\varphi}_0(t) = 1/p_0, \ p_0 = 10^6, \) and \( \tilde{\varphi}_0(t) = 0. \)

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

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

4. Update the position of each particle \( \tilde{\theta}_h(k+1) \) by (24).

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

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

7. Compute \( \tilde{\varphi}_{k+1}(t) \) by (36). Form \( \tilde{\theta}_{gl}(k+1) \) by (35), compute \( \tilde{x}_{k+1}(t) \) by (28).

(8) Compare \( \tilde{\theta}_g(k+1) \) and \( \tilde{\theta}_g(k) \): if \( \| \tilde{\theta}_g(k+1) - \tilde{\theta}_g(k) \| \leq \epsilon, \) then terminate the procedure and obtain the estimate \( \tilde{\theta}_g(k+1); \) 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 \( \xi_1, \xi_2, \) and \( \beta. \) \( \xi_1 \) and \( \xi_2 \) are positive constants between 0 and 2. \( \xi_1 \) is the step size that adjusts the particle to its own best position. \( \xi_2 \) is the step size that regulates the particle to the global best position. \( \beta \) is called the inertia factor and is an important adjusting parameter of the PSOI algorithm. A larger \( \beta \) 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. \( \xi_1 \) and \( \xi_2 \) 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 \( Y(t) \), the stacked information matrix \( \Phi(t) \), and the stacked noise vector \( V(t) \) as follows:

\[
Y(t) = [y(t), y(t-1), \ldots, y(t-p+1)]^T \in \mathbb{R}^p,
\]

\[
\Phi(t) = [\varphi(t), \varphi(t-1), \ldots, \varphi(t-p+1)]^T \in \mathbb{R}^{p \times n},
\]

\[
V(t) = [v(t), v(t-1), \ldots, v(t-p+1)]^T \in \mathbb{R}^p.
\]

From (11), we have

\[
Y(t) = \Phi(t)\theta + V(t).
\]

Define the criterion function

\[
f_1(\theta) = ||Y(t) - \Phi(t)\theta||^2.
\]

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

\[
\hat{\theta}_k(t) = \hat{\theta}_{k-1}(t) - \frac{\mu_k(t)}{2} \text{ grad } [f_1(\hat{\theta}_{k-1}(t))]
\]

\[
= \hat{\theta}_{k-1}(t) + \mu_k(t)\Theta^T(t) \left[ Y(t) - \Phi(t)\hat{\theta}_{k-1}(t) \right],
\]

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

\[
\bar{\varphi}_{1,k}(t) = \left[ -\bar{x}_{k-1}(t-1), \ldots, -\bar{x}_{k-1}(t-n_u) \right],
\]

\[
\bar{\varphi}_{2,k}(t) = \left[ \bar{x}_{k-1}(t-1), \ldots, \bar{x}_{k-1}(t-n_u) \right] \in \mathbb{R}^{n_u \times n_x},
\]

\[
\bar{\varphi}_{d,k}(t) = \left[ [\bar{v}_{k-1}(t-1), \bar{v}_{k-1}(t-2), \ldots, \bar{v}_{k-1}(t-n_d)] \right] \in \mathbb{R}^{n_d},
\]

\[
\bar{\varphi}_k(t) = \left[ \bar{\varphi}^T_{1,k}(t), \bar{\varphi}^T_{2,k}(t), \bar{\varphi}^T_{d,k}(t) \right]^T \in \mathbb{R}^n.
\]

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

\[
\bar{\Phi}_k(t) = [\bar{\varphi}_k(t), \bar{\varphi}_k(t-1), \ldots, \bar{\varphi}_k(t-p+1)]^T \in \mathbb{R}^{p \times n},
\]

and let \( \hat{\theta}_{1,k}(t) \) denote the estimates of \( \theta_1 \) at iteration \( k \)

\[
\hat{\theta}_{1,k}(t) = \left[ \bar{a}_{1,k}(t), \bar{a}_{2,k}(t), \ldots, \bar{a}_{n_u,k}(t), \bar{b}_{1,k}(t), \bar{b}_{2,k}(t), \ldots, \bar{b}_{n_x,k}(t) \right]^T \in \mathbb{R}^{n_u + n_x}.
\]

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

\[
\bar{x}_k(t) = \bar{\Phi}^T_{1,k}(t)\hat{\theta}_{1,k}(t).
\]
Compare \( \theta_k \) and \( \theta_{k-1} \): if \( \| \theta_k - \theta_{k-1} \| \leq \epsilon \), then terminate the procedure and obtain \( \theta_k \); otherwise, increase \( k \) by 1 and go to Step 2.

5. Examples

Consider the following Wiener nonlinear systems:

\[
\begin{align*}
y(t) &= f(x(t)) + D(z)v(t), \\
x(t) &= \frac{B(z)}{A(z)} u(t), \\
A(z) &= 1 + a_1z^{-1} + a_2z^{-2} = 1 - 0.43z^{-1} + 0.35z^{-2}, \\
B(z) &= b_1z^{-1} + b_2z^{-2} = 0.76z^{-1} + 0.62z^{-2}, \\
D(z) &= 1 + d_1z^{-1} + d_2z^{-2} = 1 + 0.20z^{-1} - 0.10z^{-2}, \\
f(x(t)) &= y_1x(t) + y_2x^2(t) + y_3x^3(t) \\
&= x(t) + 0.98x^2(t) + 1.15x^3(t), \\
\theta &= [a_1, a_2, b_1, b_2, y_1, y_2, y_3, d_1, d_2]^T \\
&= [-0.43, 0.35, 0.76, 0.62, 0.98, 1.15, 0.20, -0.10]^T.
\end{align*}
\]

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

<table>
<thead>
<tr>
<th>( k )</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( y_1 )</th>
<th>( y_2 )</th>
<th>( d_1 )</th>
<th>( d_2 )</th>
<th>( \delta ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.58390</td>
<td>0.68393</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.04169</td>
<td>-0.02620</td>
<td>86.40246</td>
</tr>
<tr>
<td>5</td>
<td>-0.02316</td>
<td>0.00425</td>
<td>0.61143</td>
<td>0.71531</td>
<td>0.02417</td>
<td>0.09941</td>
<td>0.13680</td>
<td>-0.03462</td>
<td>80.62844</td>
</tr>
<tr>
<td>10</td>
<td>-0.05236</td>
<td>0.01690</td>
<td>0.65537</td>
<td>0.75735</td>
<td>0.06610</td>
<td>0.28009</td>
<td>0.31492</td>
<td>-0.05635</td>
<td>72.44222</td>
</tr>
<tr>
<td>50</td>
<td>-0.19707</td>
<td>0.17193</td>
<td>0.77823</td>
<td>0.85545</td>
<td>0.55958</td>
<td>1.11096</td>
<td>0.34000</td>
<td>-0.01365</td>
<td>31.03963</td>
</tr>
<tr>
<td>100</td>
<td>-0.26799</td>
<td>0.23558</td>
<td>0.76378</td>
<td>0.81452</td>
<td>0.78248</td>
<td>1.08091</td>
<td>0.26414</td>
<td>-0.03625</td>
<td>18.93492</td>
</tr>
<tr>
<td>500</td>
<td>-0.40949</td>
<td>0.33611</td>
<td>0.75852</td>
<td>0.64487</td>
<td>0.96516</td>
<td>1.14748</td>
<td>0.20235</td>
<td>-0.07345</td>
<td>2.45484</td>
</tr>
<tr>
<td>700</td>
<td>-0.42263</td>
<td>0.34486</td>
<td>0.75695</td>
<td>0.62722</td>
<td>0.98006</td>
<td>1.16077</td>
<td>0.20205</td>
<td>-0.07432</td>
<td>1.59965</td>
</tr>
<tr>
<td>True values</td>
<td>-0.43000</td>
<td>0.35000</td>
<td>0.76000</td>
<td>0.62000</td>
<td>0.98000</td>
<td>1.15000</td>
<td>0.20000</td>
<td>-0.10000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( k )</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( y_1 )</th>
<th>( y_2 )</th>
<th>( d_1 )</th>
<th>( d_2 )</th>
<th>( \delta ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>-0.70817</td>
<td>0.59244</td>
<td>0.68145</td>
<td>0.44089</td>
<td>0.69816</td>
<td>1.06690</td>
<td>0.32058</td>
<td>-0.04582</td>
<td>27.79105</td>
</tr>
<tr>
<td>20</td>
<td>-0.74261</td>
<td>0.64034</td>
<td>0.84768</td>
<td>0.63585</td>
<td>0.98413</td>
<td>1.08805</td>
<td>0.19259</td>
<td>0.03973</td>
<td>24.33881</td>
</tr>
<tr>
<td>30</td>
<td>-0.58252</td>
<td>0.36827</td>
<td>0.80718</td>
<td>0.54617</td>
<td>0.99907</td>
<td>1.13505</td>
<td>0.10336</td>
<td>-0.10655</td>
<td>10.69988</td>
</tr>
<tr>
<td>60</td>
<td>-0.42770</td>
<td>0.35023</td>
<td>0.74459</td>
<td>0.60637</td>
<td>0.98132</td>
<td>1.15542</td>
<td>0.19161</td>
<td>-0.09934</td>
<td>1.21353</td>
</tr>
<tr>
<td>90</td>
<td>-0.43027</td>
<td>0.35147</td>
<td>0.75416</td>
<td>0.61506</td>
<td>0.97971</td>
<td>1.15157</td>
<td>0.19172</td>
<td>-0.09235</td>
<td>0.72682</td>
</tr>
<tr>
<td>120</td>
<td>-0.43132</td>
<td>0.35162</td>
<td>0.75594</td>
<td>0.61617</td>
<td>0.97964</td>
<td>1.15089</td>
<td>0.19158</td>
<td>-0.09109</td>
<td>0.71978</td>
</tr>
<tr>
<td>True values</td>
<td>-0.43000</td>
<td>0.35000</td>
<td>0.76000</td>
<td>0.62000</td>
<td>0.98000</td>
<td>1.15000</td>
<td>0.20000</td>
<td>-0.10000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

Figure 3: The GI estimation errors versus \( k \).

Figure 4: The PSOI estimation errors versus \( k \).
simulation, each swarm generation contains 10 position particles. The coefficients $\beta$, $\xi_1$, and $\xi_2$ 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


