Intelligent Automation And Soft Computing, 2019
Copyright © 2019, TSI® Press
Vol. 25, no. 2, 329–341
https://doi.org/10.31209/2019.100000093

Modified PSO Algorithm on Recurrent Fuzzy Neural Network for System Identification

Chung Wen Hung, Wei Lung Mao, Han Yi Huang

Graduate School of Engineering Science and Technology and Department of Electrical Engineering, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan

ABSTRACT

Nonlinear system modeling and identification is the one of the most important areas in engineering problem. The paper presents the recurrent fuzzy neural network (RFNN) trained by modified particle swarm optimization (MPSO) methods for identifying the dynamic systems and chaotic observation prediction. The proposed MPSO algorithms mainly modify the calculation formulas of inertia weights. Two MPSOs, namely linear decreasing particle swarm optimization (LDPSO) and adaptive particle swarm optimization (APSO) are developed to enhance the convergence behavior in learning process. The RFNN uses MPSO based method to tune the parameters of the membership functions, and it uses gradient descent (GD) based scheme to optimize the parameters of the conclusion part of the fuzzy system. The effectiveness of our method is evaluated for three nonlinear system modelling and signal prediction, including Henon system, nonlinear plant system and Mackey-Glass time series. Simulation results show that the proposed RFNN with LDPSO algorithm can provide more effective and accurate identification performances compared with the APSO method in term of mean squared error (MSE).

KEYWORDS: Recurrent fuzzy neural network (RFNN), modified particle swarm optimization (MPSO), gradient descent (GD) algorithm, dynamic system identification.

1 INTRODUCTION

A system model is the mathematical description of a physical, biological, or information system. By using system identification process, a compact and accurate mathematical model of a dynamic system can be found based on experimental data. It is one of the most important areas in controller design, optimization, fault detection, and system engineering [1, 2] because of its applicability to a wide range of problems. Typically, the physical models are nonlinear systems where the parameter variations are unavoidable. It is necessary to develop the efficient and high accuracy model estimation approach using suitable and simple techniques for different system structures.

Nonlinear and complex dynamic system identification has attracted considerable research attention, and many mathematical models are analyzed and implemented from given input–output measurements. Several modern computing approaches such as artificial neural network model [8-18], support vector machines [4-6], genetic algorithm [7], and particle swarm algorithm [3, 17-31] are applied in system modeling. The conventional support vector machine (SVM) method [6] presents the linear and nonlinear system identification applications. It offers the advantage that the number of kernels and center parameters are found automatically. The least square support vector machine (LS-SVM) [4, 5] has been applied in function estimation problems and is used for modeling of multi-input multi-output (MIMO) Hammerstein autoregressive with exogenous terms (ARX) systems. This approach can determine the memoryless static nonlinearity as well as the linear model parameters from a linear set of equations. However, the method of automatic determination of the order of process model still has not been established.

In effort to resolve this problem, some artificial intelligent and soft computing approaches are recently introduced [8-13], such as fuzzy systems, neural networks, and particle swarm algorithm [15] applied in system identification. The adaptive-network-based fuzzy inference system (ANFIS) [9, 10] with

CONTACT Chung Wen Hung wenhung@yuntech.edu.tw

backpropagation training is adapted for nonlinear system modeling problem. The fuzzy neural network (FNN) [11, 12] has become a popular structure for system modeling because it incorporates both of their advantages. The existing forward FNN is limited to static problems due to their feedforward structure, which causes the inefficiency for temporal problems. So, the recurrent fuzzy neural networks [12-16] are proposed to solve this difficulty.

The recurrent fuzzy neural network (RFNN) [13] based on supervised learning is used to identify and control a nonlinear dynamic system. It uses a dynamic mapping network and is more suitable for representing dynamic systems than the conventional FNN. The learning algorithm based on the gradient method is presented for control and identification. Many different optimization methods are used to train the parameters of fuzzy systems and neural networks. These algorithms can be classified as derivative-free and derivative-based optimization methods. Thus, the gradient method suffers from slow convergence and stability, and it cannot obtain a better approach during the learning process. The hybrid learning algorithm can adapt the network parameters to increase its performance [14-16]. In [14], Takagi-Sugeno-Kang (TSK) type recurrent fuzzy network (TRFN) are designed and trained by hybrid of a multi-group GA and particle swarm optimization (R-MGAPSO) algorithms, whose performance is then shown to be superior to that of GA method for nonlinear control of dynamic plant. Due to the computational simplicity and search good solution rapidly, PSO-based algorithm [15] is employed for online parameter identification. An improve PSO algorithm [31] which uses the dynamic decreasing inertia weight is presented to identify the Hammerstein model. The dynamic inertial weight scheme is designed to increase the convergence speed and accuracy during the learning process.

In this paper, the hybrid algorithm composed of modified particle swarm optimization and gradient descent algorithm is presented for recurrent fuzzy neural network in the application of system modeling. The MPSO algorithms with linear decreasing inertia weight or adaptive inertia weight are proposed to train the parameters of the antecedent part of the fuzzy system. A large inertia weight has extensive search ability while a small inertia weight is beneficial to local search and to obtain more precise solution, but sometimes it is easy to fall into local optimum. The MPSO scheme can achieve better performance in terms of convergence rate and quality of solution. The GD algorithm is applied to tune the parameters of conclusion part. The proposed algorithm can offer the recursive propagation and network weight computation, and it also provides faster convergence and tracking capabilities for nonlinear system identification. Three kinds of system are considered, namely, (1) Henon system, (2) nonlinear system, and (3) Mackey-Glass time series. The simulation results of the RFNN trained by hybrid algorithm are obtained to verify the performances in terms of mean squared error (MSE) under different nonlinear system conditions.

This paper is organized as follows. The system model problem is introduced in section 2. The recurrent fuzzy neural network structure is described in section 3. Section 4 presents the proposed MPSO algorithm and GD scheme for the various system modeling. Section 5 provides the simulation results. Lastly, briefly conclusion is given in section 6.

2 SYSTEM IDENTIFICATION PROBLEM

SYSTEM identification is a way to build a mathematical model of system behavior based on the relationship between the input and output measurement data. The main purpose hopes that through the measured input and output information to predict the evolution of the output of the system in the future. System identification was also used in various fields widely, such as stock forecasting or control theory, etc. Before analyzing the unknown system, it is assumed that there is a n-dimensional nonlinear system which can be represented as follow:

$$\dot{\mathbf{X}}_s = \mathrm{F}(\mathbf{X}_s, \mathbf{X}_0, \boldsymbol{\theta}) \tag{1}$$

where $\mathbf{X}_s = [x_1, x_2, \ldots, x_n]^T \in R^n$ is the system state vector, $\mathbf{X}_0$ denotes the initial system state, $\theta = [\theta_1, \theta_2, \ldots, \theta_n]^T \in R^n$ is the system parameter vector, and F is a nonlinear system function. In order to estimate the unknown system parameters, the system identification can be defined as：

$$\dot{\hat{\mathbf{X}}}_s = \mathrm{F}(\hat{\mathbf{X}}_s, \mathbf{X}_0, \hat{\boldsymbol{\theta}}) \tag{2}$$

where $\hat{\mathbf{X}}_s = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n]^T$ is the estimated system state vector, and $\hat{\theta} = [\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n]^T$ is the estimated system parameter vector.

In this study, an objective function needs to be first defined. The mean squared error (MSE) between actual and estimated value of the output as objective function value can be considered as a fitness function. The objective function is defined as follows：

$$\mathrm{MSE} = \frac{1}{N}\sum_{k=1}^{N}\left\| X_k - \hat{X}_k \right\| \tag{3}$$

where N is the length of the data used for system identification, X_k is the actual value of the output, and $\hat{X}_k$ is the estimated value of the output.

3 RECURRENT FUZZY NEURAL NETWORK MODEL

A type of the recurrent fuzzy neural network is utilized here. The proposed structure is derived from ANFIS structure. In this structure, each membership function (MF) has a feedback connection which can memorize the past information to establish the temporal relations. On the other hand, the whole architecture is based on the fuzzy inference system as the main body to enhance the processing capabilities of uncertainty and imprecisely for the system. Moreover, fuzzy inference system also combines with the property of self-learning of the neural network to adjust parameters. A schematic diagram of the RFNN structure is shown in Figure 1. The model in this way is consisted of six layers. It is assumed that there are M input nodes, and the inputs can be represented by a M dimensional vector $\mathbf{X} = [x_1, x_2, \ldots, x_M]$.

Layer 0 represents to transmit the input values to the next layer directly. The outputs of this layer can be written as：

$$O_i^0(k) = x_i^0(k) \qquad i = 1, 2, \ldots, M \tag{4}$$

where the superscript indicates the layer number.

Layer 1 denotes fuzzification layer that makes the input variables map to the fuzzy sets by Gaussian membership functions. This layer performs the fuzzification process of the RFNN. The relationship between input and output is as follows：

$$O_{ij}^1(k) = \exp\left\{-\frac{\left(u_{ij}^1(k) - m_{ij}(k)\right)^2}{\sigma_{ij}^2(k)}\right\} \qquad j = 1, 2, \ldots, N \tag{5}$$

where $m_{ij}(k)$ and $\sigma_{ij}(k)$ are the center point and the standard deviation of the j-th Gaussian MF of the i-th input respectively. In addition, for the discrete time k, the inputs which contain feedback connections in this layer are denoted by：

$$u_{ij}^1(k) = O_i^0(k) + O_{ij}^f(k) \tag{6}$$

where $O_{ij}^f(k)$ is defined as：

$$O_{ij}^f(k) = \theta_{ij}(k) O_{ij}^1(k-1) \tag{7}$$

where $\theta_{ij}(k)$ is the feedback weight. The added feedback connections to a static ANFIS structure can make it become a dynamic RFNN structure. Each node in this layer has three adjustable parameters, those are parameters m_{ij}, σ_{ij}, and θ_{ij}.

Layer 2 denotes the rule layer that makes the fuzzy sets of each input variable execute permutation and combination. At this layer, the nodes represent fuzzy rules and the outputs utilize fuzzy “AND” operation to compute the “firing strength” of corresponding rule. Where the “AND” operation is represented by multiplication of corresponding MFs.

Layer 3 denotes normalization layer that makes the rule of each node in second layer perform normalization. The outputs of this layer can be expressed as：

$$O_p^3(k) = \frac{O_p^2(k)}{\sum_{p=1}^{N^M} O_p^2(k)} \qquad p = 1, 2, \ldots, N^M \tag{8}$$

where N^M is the number of rules or nodes in second layer.

Layer 4 denotes conclusion inference layer that makes the normalized rules in layer 3 multiply by corresponding TSK fuzzy model. The outputs of this layer can be expressed as：

$$O_p^4(k) = O_p^3(k) F_p(k) \tag{9}$$

where $F_p(k)$ is the TSK fuzzy model which considered here are linear combinations of the current inputs of the system plus a constant. It can be written as：

Rule p: If x_1 is A_{p1} and x_2 is A_{p2} and … x_M is A_{pM} then

$$F_p(k) = \alpha_{p0}(k) + \alpha_{p1}(k) x_1 + \ldots + \alpha_{pM}(k) x_M \tag{10}$$

where the coefficients α are the adjustable parameters.

Layer 5 denotes output layer that performs summation of each output in layer 4. Finally, the output of the RFNN can be written as：

$$y(k) = O_1^5(k) = \sum_{p=1}^{N^M} O_p^4(k) \tag{11}$$

The RFNN structure [13] can be shown to be a universal approximation for continuous function over compact sets.

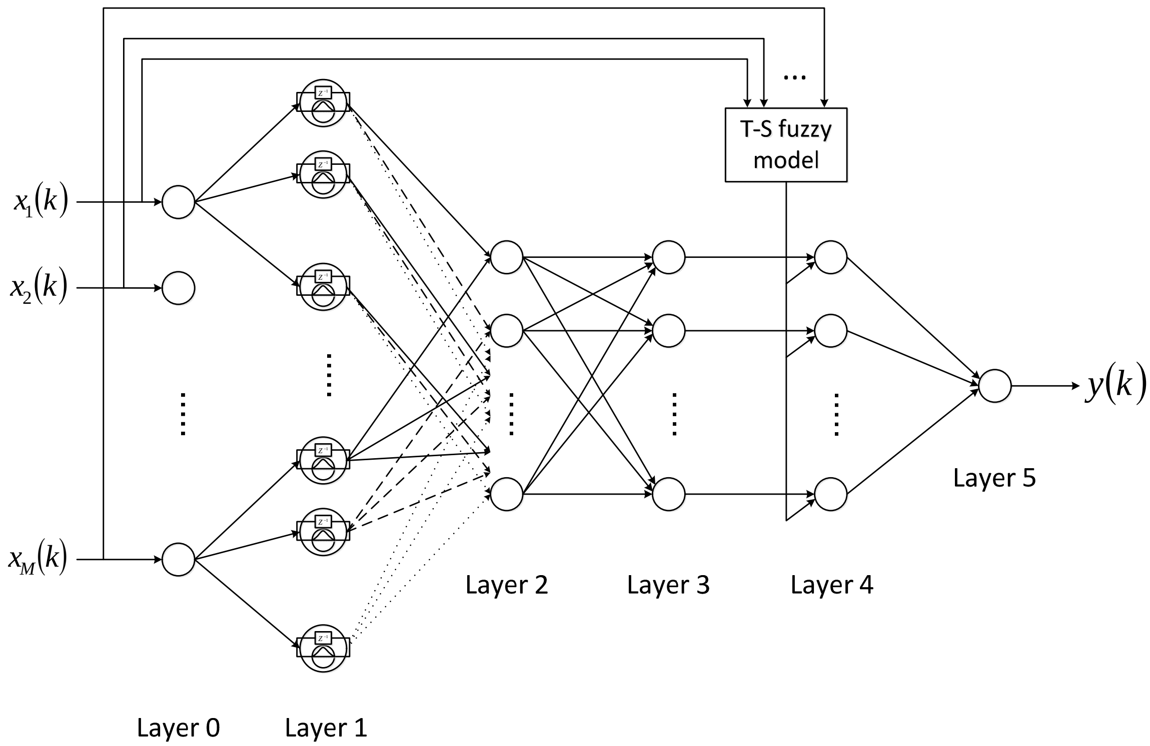


Figure 1 The structure of the RFNN

4 MODIFIED PARTICLE SWARM OPTIMIZATION (MPSO)

PARTICLE swarm optimization (PSO) which is originally proposed by Kennedy and Eberhart [26] is a bionic technique based on swarm intelligence. Two scholars got inspiration by observing the behavior of birds feeding. The PSO is a stochastic optimization algorithm. In the population, each particle is an independent individual, and the swarm is composed of particles. Through rules of interaction between the individual and the individual produce specific group behavior to achieve the goal of optimization.

4.1 Standard PSO algorithm

In the theory of PSO algorithm, the problem solution space is formulated as a search space. Each particle in the search space is simulated as a bird, and the position of each particle represents a potential solution of the problem. In addition, the particles fly in a multi-dimensional search space. In order to search for the optimal solution, the PSO also develops a way to update velocity which can decide the movement direction and distance of each particle. Each particle updates its velocity based on the current velocity and the best position experienced by particle and the best position experienced by swarm. Moreover, utilizing an objective function value or a fitness value to determine the solution is good or bad.

Assume that there is a d-dimensional search space. In this search space, each particle is associated with the velocity vector $\mathbf{V}_i = [v_i^1, v_i^2, \ldots, v_i^d]$ and the position vector $\mathbf{X}_i = [x_i^1, x_i^2, \ldots, x_i^d]$, where the subscript i indicates the i-th particle of the swarm. Moreover, the best position which has experienced by particle can be expressed as $\mathbf{P}_i = [p_i^1, p_i^2, \ldots, p_i^d]$ and the best position which has experienced by the whole swarm is $\mathbf{P}_g = [p_g^1, p_g^2, \ldots, p_g^d]$. According to the previous definition, the velocity and position of each particle can be updated by the following equations：

$$v_i(t+1) = wv_i(t) + c_1r_1(p_i(t) - x_i(t)) + c_2r_2(p_g(t) - x_i(t)) \tag{12}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{13}$$

where t is the current number of iterations. c_1 and c_2 are two positive constants which are normally taken value in interval $[0, 2]$. r_1 and r_2 are two uniformly distributed numbers between 0 and 1. Also, w is the inertia weight which can develop the search space. Besides, in every iteration, the particle best position $\mathbf{P}_i$ and global best position $\mathbf{P}_g$ also need to be updated by following conditions：

$$\mathbf{P}_i(t+1)=\begin{cases}\mathbf{P}_i(t), & f(\mathbf{X}_i(t+1))\geq f(\mathbf{P}_i(t))\\ \mathbf{X}_i(t+1), & f(\mathbf{X}_i(t+1))< f(\mathbf{P}_i(t))\end{cases} \tag{14}$$

$$\mathbf{P}_g(t)\in\{\mathbf{P}_0(t),\mathbf{P}_1(t),\ldots,\mathbf{P}_s(t)\}\,|\,f(\mathbf{P}_g(t))=\min(f(\mathbf{P}_0(t)),f(\mathbf{P}_1(t)),\ldots,f(\mathbf{P}_s(t))) \tag{15}$$

where f is the objective function or fitness function, and s is the total number of particles.

4.2 Linear decreasing PSO (LDPSO) algorithm

The inertia weight plays an important role in the convergence of the PSO algorithm. The main problem of using the inertia weight is how to set a reasonable value so that the particles can obtain a better balance between the global search and local search. The inertia weight was originally designed as a constant. In order to improve the convergence property of the PSO algorithm, a linear decreasing inertia weight is applied as follow:

$$w(t)=w_{max}-(w_{max}-w_{min})\frac{t}{t_{max}} \tag{16}$$

where t is the current number of iterations, t_{max} is the maximal number of iterations. w_{max} and w_{min} are the maximal and minimal weights, respectively. The main concept of linear decreasing is to give larger inertia weight during the early search, so that the particles can keep larger velocity to perform global search. Subsequently, with the increase of iterations, the inertia weight will decline and cause the velocity of particles slow down to perform local search.

4.3 Adaptive PSO (APSO) algorithm

In this section, an adaptive inertia weight adjustment strategy is introduced. This method must determine the updated situation of particles at each iteration. In this method, the success of the particle is defined as follow:

$$S_i(t)=\begin{cases}1 & \text{if } f(\mathbf{P}_i(t))< f(\mathbf{P}_i(t-1))\\ 0 & \text{if } f(\mathbf{P}_i(t))= f(\mathbf{P}_i(t-1))\end{cases} \tag{17}$$

where $\mathbf{P}_i(t)$ is the best position which has experienced by particle i until t-th iteration, and f is the objective function or fitness function. Also, the success percentage of the swarm is defined as：

$$P_s(t)=\frac{\sum_{i=1}^{n}S_i(t)}{n} \tag{18}$$

where n is the number of particles, and P_s belongs to the interval $[0,1]$. A high percentage of success expresses that the particles located to a point which is far from the optimum point and moved slowly toward the optimum. Also, a low percentage of success indicates that the particles oscillated around the optimum without much improvement.

Therefore, the adaptive inertia weight which uses a linear function to map the P_s can be written as：

$$w(t)=(w_{max}-w_{min})P_s(t)+w_{min} \tag{19}$$

where the range of the inertia weight $[w_{min},w_{max}]$ is usually set as $[0,1]$.

5 HYBRID LEARNING ALGORITHM FOR TRAINING RFNN

IN this section, a hybrid learning algorithm which is composed of gradient descent (GD) and modified particle swarm optimization (MPSO) for training RFNN model is discussed. This learning algorithm of the RFNN uses MPSO based method to optimize the parameters of the antecedent part in fuzzy system and GD based method to optimize the parameters of the conclusion part. In antecedent part of the RFNN, there are three parameters must be trained for each membership function (MF). These parameters are the center point m and standard deviation σ of MF, and the weight θ of feedback connection which can supply the past information. We utilize MPSO based method to adjust these parameters. In population, the elements of each particle position mainly consist of the parameters, those are mean (m), spread (σ), and feedback weight (θ). Therefore, the position vector of the i-th particle can be expressed as:

$$\mathbf{X}_i=[m_i^1,m_i^2,\ldots,m_i^g,\sigma_i^1,\sigma_i^2,\ldots,\sigma_i^g,\theta_i^1,\theta_i^2,\ldots,\theta_i^g] \tag{20}$$

where g is the number of MFs. Then the particles can update their respective position by the Equations (12-19) to optimize these parameters.

In the conclusion part of the RFNN, GD based method is used to optimize the coefficients α of the TSK fuzzy model. Assumed that the cost function is expressed as follow：

$$E(k)=\frac{1}{2}\sum_{k=1}^{N}(T(k)-O_1^5(k))^2 \tag{21}$$

where $T(k)$ is the target value of k-th sample. Using the chain rule for GD based method to verify the update of the coefficients α is developed as follows：

$$\Delta\alpha_{p0}(k) = -\eta\frac{\partial E(k)}{\partial\alpha_{p0}(k)} = -\eta\frac{\partial E(k)}{\partial O_1^5(k)}\frac{\partial O_1^5(k)}{\partial O_p^4(k)}\frac{\partial O_p^4(k)}{\partial\alpha_{p0}(k)} = \eta\, e(k) O_p^3(k) \tag{22}$$

$$\Delta\alpha_{pi}(k) = -\eta\frac{\partial E(k)}{\partial\alpha_{pi}(k)} = -\eta\frac{\partial E(k)}{\partial O_1^5(k)}\frac{\partial O_1^5(k)}{\partial O_p^4(k)}\frac{\partial O_p^4(k)}{\partial\alpha_{pi}(k)} = \eta\, e(k) O_p^3(k)\cdot x_i\ ,\ i \neq 0 \tag{23}$$

with $e(k) = T(k) - O_1^5(k)$

where x_i is the i-th input of the RFNN, and η is the learning rate.

In our method, the MPSO method can update the RFNN membership function parameters, and the GD method can tune the consequent parameters using the training data information. Then, the MSE of the training pattern is calculated after the consequent and antecedent parameters are updated. The iteration process is operated to minimize the MSE value.

6 SIMULATION RESULTS

IN this section, we evaluated the performance of the RFNN with MPSO-GD learning algorithm for temporal problems. Three simulation results are illustrated to compare the LDPSO method and APSO method. The proposed prediction system is developed to examine the Henon system, nonlinear plant, and Mackey-Glass. The population size of the MPSO for all of the experiments is set as 10, and the parameters $c_1 = 1$ and $c_2 = 1$ are selected. Also the velocity limit $[v_{min}, v_{max}]$ of the MPSO is $[-0.1, 0.1]$. Moreover the learning rate η is chosen as 0.15. The maximum number of iterations is 500. These parameters are determined by empirical rules to achieve a better MSE in the experimentation conditions. Table 1 and Table 2 summarize the comparison MSE performances with respect to the three identification problem.

A. Henon system identification

Henon system is a second order time delay difference equation that presents chaotic characteristics. This chaotic system which has two parameters and one delay is defined as follow:

$$y(k+1) = -P\cdot y^2(k) + Q\cdot y(k-1) + 1.0 \tag{24}$$

where the parameters are $P = 1.4$, and $Q = 0.3$. The initial values are $y(1) = 0.4$, and $y(0) = 0.4$. Comparisons of simulation results of LDPSO and APSO algorithms are shown in Figure 2. We choose 1000 training data and 1000 test data. The inputs of the RFNN are chosen as $[y(k) \quad y(k-1)]$, and each input has four membership functions. In the training period, the situation of the MSE convergence and inertia weight variation of the LDPSO and APSO methods are shown in Figure 2(a) and (b). The original data samples of points 1–1000 are used as the training dataset, and the prediction results of the LDPSO and APSO methods are illustrated in Figure 2(c). The error of the prediction results of training dataset based on the LDPSO and APSO methods are shown in Figure 2(d). Also samples 1001 to 2000 are the checking dataset for validation. The prediction results of the LDPSO and APSO methods are illustrated in Figure 2(e). The error of the prediction results of checking dataset based on the LDPSO and APSO methods are shown in Figure 2(f). On average, the MSE of LDPSO method for training data is 2.552×10^{-5} and for validation data is 8.359×10^{-5} while the MSE of APSO method for training data is 5.155×10^{-5} , and for validation data is 1.249×10^{-4} . The LDPSO-based method provides the better convergence performance for RFNN state estimation.

B. Nonlinear plant system identification

In this example, a nonlinear plant with multiple time delays is defined by the following difference equation：

$$y_p(k+1) = f\left(y_p(k), y_p(k-1), y_p(k-2), u_p(k), u_p(k-1)\right) \tag{25}$$

with

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1x_2x_3x_5(x_3-1) + x_4}{1 + x_2^2 + x_3^2}$$

where the initial values are $y_p(0) = 0$, and $u_p(0) = 0$. The current output of the nonlinear plant relates to three previous outputs and two previous inputs. The inputs of the RFNN are chosen as $[u_p(k) \quad y_p(k)]$ and each input has four membership functions. We chose 900 training data to train the RFNN, and the equation is shown in below：

$$u_p(k) = \begin{cases} \text{distributed uniform over } [-2,2] & 0 < k \le 450 \\ 1.05\sin\left(\dfrac{\pi k}{45}\right) & 450 < k \le 900 \end{cases} \tag{26}$$

There is no repetition of these 900 training data. On the other hand, the 1000 checking data are generated to verify the estimation results by the following equation. That is

Table 1 The MSE performances of training data for three different systems.

	Henon system		Nonlinear plant model		Mackey-Glass system	
	LDPSO	APSO	LDPSO	APSO	LDPSO	APSO
1	4.186e-05	4.436e-05	5.230e-02	5.890e-02	1.184e-04	1.469e-04
2	3.999e-05	5.685e-05	5.020e-02	5.180e-02	1.478e-04	1.751e-04
3	5.126e-06	6.087e-05	4.600e-02	4.790e-02	9.282e-05	1.369e-04
4	2.859e-05	7.905e-05	5.090e-02	5.380e-02	1.029e-04	1.654e-04
5	2.141e-05	5.506e-05	5.320e-02	5.850e-02	1.203e-04	1.658e-04
6	3.495e-05	4.516e-05	4.900e-02	5.780e-02	1.797e-04	1.742e-04
7	1.661e-05	4.485e-05	5.060e-02	5.580e-02	1.684e-04	1.856e-04
8	2.383e-05	4.436e-05	4.970e-02	5.350e-02	1.217e-04	1.593e-04
9	1.424e-05	5.926e-06	5.260e-02	5.400e-02	1.453e-04	1.955e-04
10	2.859e-05	7.905e-05	5.330e-02	5.880e-02	1.184e-04	1.469e-04
Averaged MSE	2.552e-05	5.155e-05	5.078e-02	5.508e-02	1.316e-04	1.652e-04

Table 2 The MSE performances of validation results for three different systems.

	Henon system		Nonlinear plant model		Mackey-Glass system	
	LDPSO	APSO	LDPSO	APSO	LDPSO	APSO
1	5.191e-05	7.520e-05	3.800e-03	3.300e-03	8.152e-05	1.799e-04
2	5.557e-05	6.650e-05	2.500e-03	1.900e-03	8.996e-05	1.088e-04
3	1.208e-05	2.047e-04	2.700e-03	2.200e-03	7.412e-05	8.222e-05
4	3.226e-05	7.120e-05	3.000e-03	2.500e-03	7.152e-05	1.077e-04
5	2.915e-04	6.740e-05	3.240e-02	2.700e-03	7.767e-05	1.143e-04
6	7.386e-05	2.597e-04	3.300e-03	2.200e-03	1.327e-04	1.176e-04
7	2.161e-04	3.124e-04	2.500e-03	3.100e-03	1.668e-04	1.160e-04
8	5.191e-05	7.520e-05	2.200e-03	2.000e-03	7.425e-05	8.764e-05
9	1.844e-05	4.607e-05	2.000e-03	2.600e-03	1.077e-04	1.361e-04
10	3.226e-05	7.120e-05	6.800e-03	2.300e-03	8.152e-05	1.799e-04
Averaged MSE	8.359e-05	1.249e-04	6.120e-03	2.480e-03	9.576e-05	1.230e-04

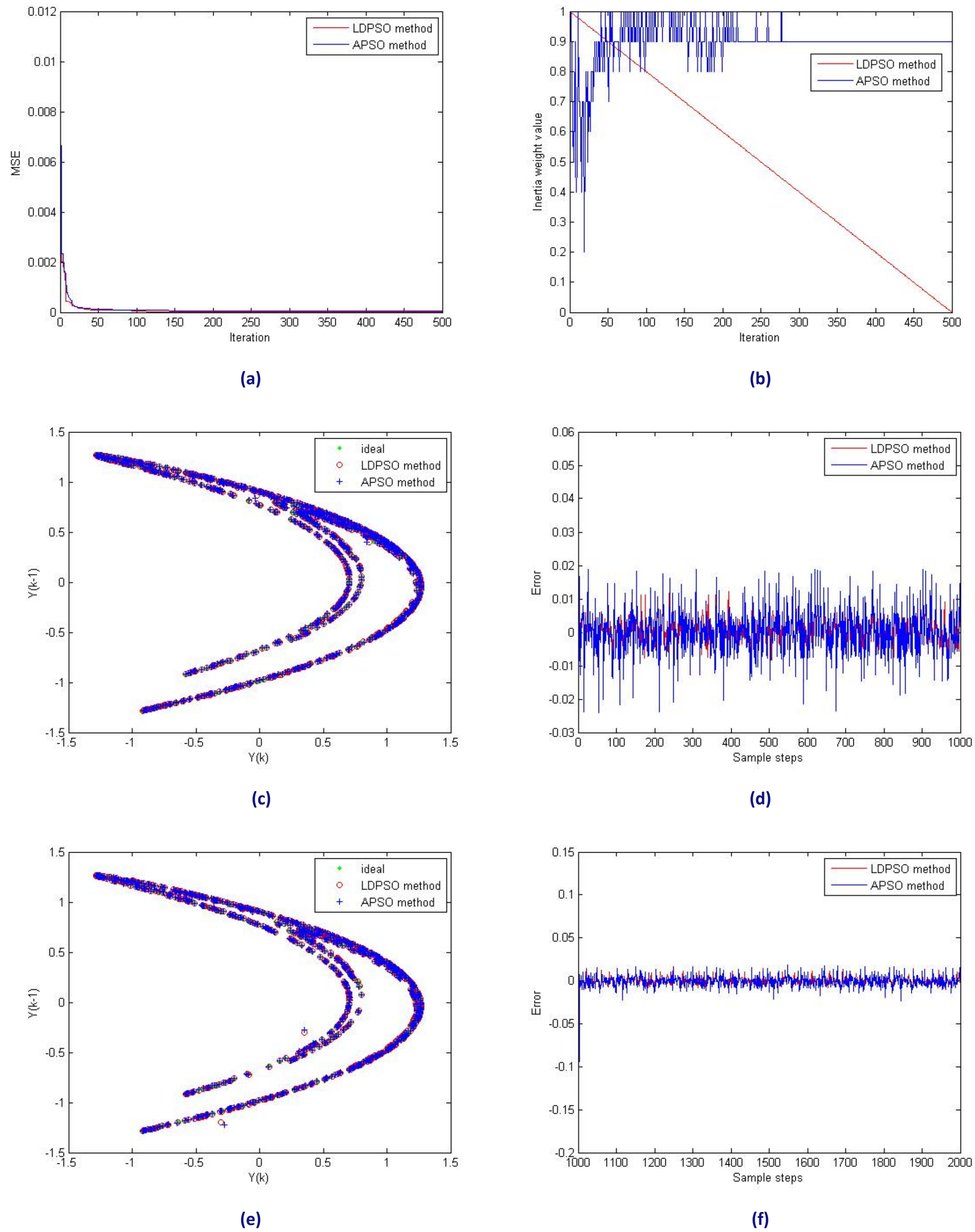


Figure 2 **Henon system identification (a) the convergence curve for the two kinds of methods, (b) inertia weight of LDPSO and APSO methods applied to Henon system, (c) results of identification of train data for Henon system time series, (d) the error results of training data for the two kinds of methods, (e) results of identification of test data for Henon system time series, (f) the error results of test data for the two kinds of methods.**

$$u_p(k)=\begin{cases}\sin\left(\frac{\pi k}{25}\right) & 0<k\le 250\\ 1.0 & 250<k\le 500\\ -1.0 & 500<k\le 750\\ 0.3\sin\left(\frac{\pi k}{25}\right)+0.1\sin\left(\frac{\pi k}{32}\right)+0.6\sin\left(\frac{\pi k}{10}\right) & 750<k\le 1000\end{cases} \tag{27}$$

Comparisons of simulation results of LDPSO and APSO algorithms are shown in Figure 3. In the training period, the situation of the MSE convergence and inertia weight variation of the LDPSO and APSO methods are shown in Figure 3(a) and (b). For the training dataset, the prediction results of the LDPSO and APSO methods are illustrated in Figure 3(c). The error of the prediction results of training dataset based on the LDPSO and APSO methods are shown in Figure 3(d). For the checking dataset, the prediction results of the LDPSO and APSO methods are illustrated in Figure 3(e). The error of the prediction results of checking dataset based on the LDPSO and APSO methods are shown in Figure 3(f). On average, the MSE of LDPSO method for training data is 0.05078 and for test data is 0.00612 while the MSE of APSO method for training data is 0.05508 and for test data is 0.00248. The LDPSO based method provides the better convergence and accuracy for nonlinear system modelling.

C. Mackey-Glass system

Mackey-Glass equation is a nonlinear time delay differential equation that presents chaotic characteristics. It is originally proposed to model the physiological signals generated and sensed from a nonlinear system. The system can be described by the following equation：

$$\dot{x}(k)=\frac{\alpha x(k-\tau)}{1+x^{\lambda}(k-\tau)}-\beta x(k) \tag{28}$$

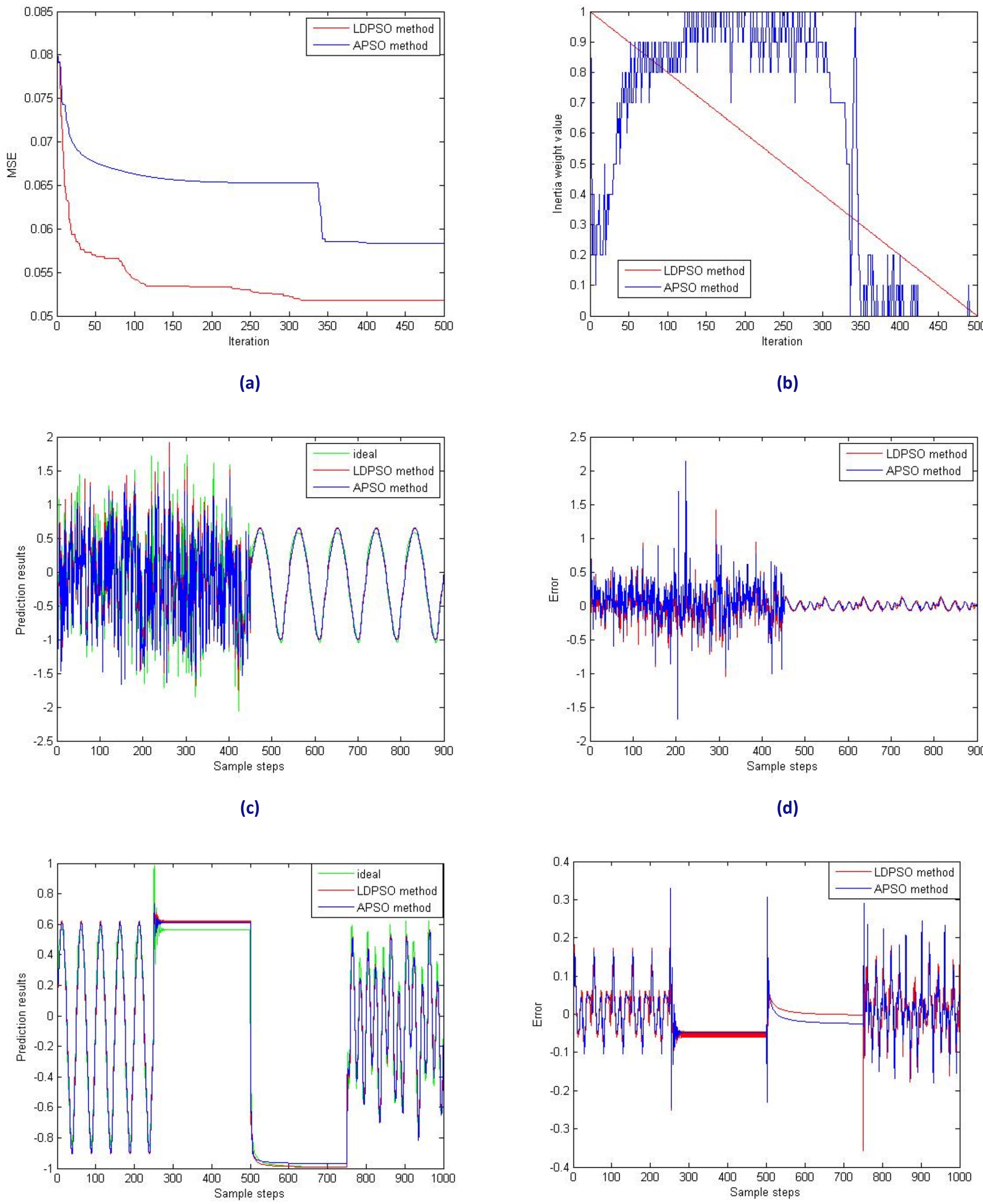
where the parameters are $\alpha=0.2$, $\beta=0.1$, $\lambda=10$, and $\tau=17$. One thousand data samples are generated with an initial value of $x(0)=0.1$ based on numerical integration using fourth-order Runge–Kutta method and a time step of 0.1.

Comparisons of simulation results of LDPSO and APSO algorithms are shown in Figure 4. We chose 600 training data and 400 test data. The inputs of the RFNN are chosen as $\left[x(k) \quad x(k-1) \quad x(k-2) \quad x(k-3)\right]$, and each input has four membership functions. In the training period, the situation of the MSE convergence and inertia weight variation of the LDPSO and APSO methods are shown in Figure 4(a) and (b). The original data samples of points 1–600 are used as the training dataset, and the prediction results of the LDPSO and APSO methods are illustrated in Figure 4(c). The error of the prediction results of training dataset based on the LDPSO and APSO methods are shown in Figure 4(d). Also samples 601 to 1000 are the checking dataset for validation. The prediction results of the LDPSO and APSO methods are illustrated in Figure 4(e). The error of the prediction results of checking dataset based on the LDPSO and APSO methods are shown in Figure 4(f). On average, the MSE of LDPSO method for training data is 1.316×10^{-4} and for test data is 9.576×10^{-5} while the MSE of APSO method for training data is 1.652×10^{-4} and for validation data is 1.230×10^{-4}. Also, the MPSO-GD RFNN structure can achieve the better convergence performances for signal prediction and modeling.

In this research, we have chosen the MSE as the criterion of performance evaluation. The MSE of all experimental results are tabulated in Table 1 (training results) and Table 2 (validation results). The predictions of averaged MSE of the nonlinear system models which are divided into LDPSO and APSO methods are simulated and computed by 10 runs, respectively. It is shown that the LDPSO method outperforms APSO method for modelling of Mackey-Glass chaotic system.

7 CONCLUSIONS

THIS research presents an RFNN with hybrid training algorithm for system modeling and identification of measurement data. The recurrent structure RFNN with university approximation property is developed to deal with the temporal problem. The LDPSO and APSO schemes with inertia weight adjustment are investigated to train the parameters of the antecedent part of the fuzzy membership function. With the use of MPSO-GD algorithm, the weight parameters are tuned to optimize the MSE metric. The modified recursions are derived and the corresponding identification performances for three types of dynamic system datasets, which are Henon system, nonlinear plant, and Mackey-Glass time series, are tested. The MPSO schemes can achieve superior accuracy to robustly identify the nonlinear system and time series signals. On average, the proposed LDPSO-GD RFNN offers the better MSE performances and faster adaptation capability than the APSO-GD method in the nonlinear system modeling problem. This research can establish a good promising MPSO RFNN for online system identification. This method can set the path for more research related to identification problem, and it can be developed and implemented to be a useful PSO algorithm for industry applications in the future.



(e) (f)

Figure 3 Nonlinear plant system identification (a) the convergence curve for the two kinds of methods, (b) inertia weight of LDPSO and APSO methods applied to nonlinear plant, (c) results of identification of train data for nonlinear plant time series, (d) the error results of train data for the two kinds of methods, (e) result of identification of test data for nonlinear plant time series, (f) the error results of test data for the two kinds of methods.

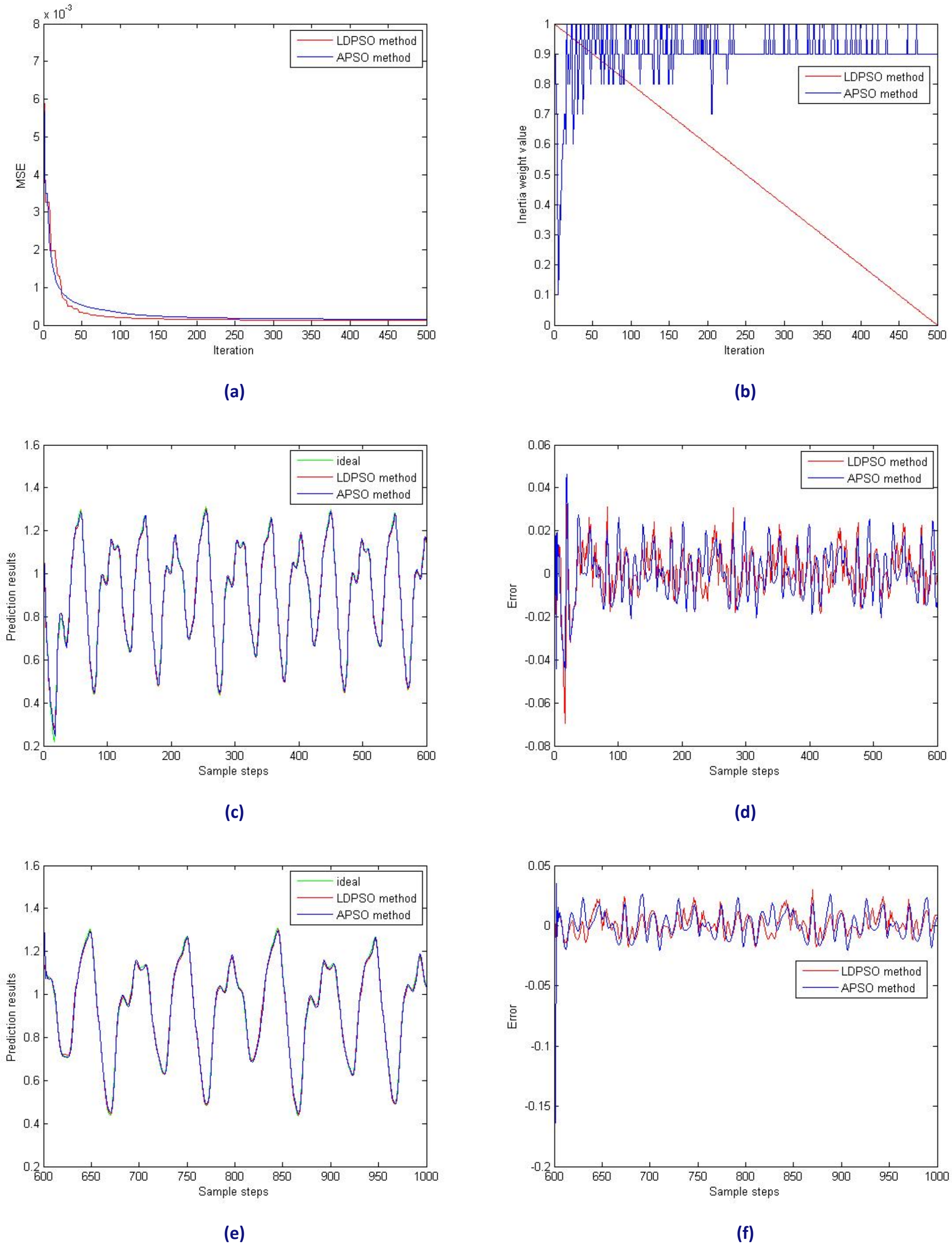


Figure 4 Mackey-Glass chaotic signal prediction (a) the convergence curve for the two kinds of methods, (b) inertia weight of LDPSO and APSO methods applied to Mackey-Glass, (c) results of identification of train data for Mackey-Glass time series, (d) the error results of train data for the two kinds of methods, (e) result of identification of test data for Mackey-Glass time series, (f) the error results of test data for the two kinds of methods.

8 ACKNOWLEDGMENT

THE authors would like to thank the Ministry of Science and Technology of the Republic of China, Taiwan, for financially supporting this research under Contract No. MOST 105-2622-E-224 -010 –CC3 and MOST 106-2221-E-224-025-.

9 REFERENCES

Alfi, A. (2012). Particle swarm optimization algorithm with dynamic inertia weight for online parameter identification applied to Lorenz chaotic system. ICIC International Journal of Innovative Computing, Information and Control, 8(2), 1191-1203.

Bansal, J. C., Singh, P. K., Saraswat, M., Verma, A., Jadon, S. S., Abraham, A. (2011, Nov.). Inertia weight strategies in particle swarm optimization. 2011 third world congress on nature and biologically Inspired Computing (pp. 633-640). Salamanca, Spain: IEEE.

Chen, D. W., Zhang, J. P. (2005, Aug.). Time series prediction based on ensemble ANFIS. IEEE International Conference on Machine Learning and Cybernetics (pp. 3552-3556), Guangzhou, China: IEEE.

Drezet, P. M. L., Harrison, R. F. (1998, Sep.). Support vector machines for system identification. International Conference on Control (pp. 688-692). Swansea, UK: IET.

Engelbrecht, A. P. (2005). Fundamentals of computational swarm intelligence. Wiley.

Fu, L., Li, P. (2003, Aug.). The Research Survey of System Identification Method. International conference on intelligent human-machine systems and cybernetics (pp. 397–401). Hangzhou, China: IEEE.

Ge, H. W., Liang, Y. C., & Marchese M. (2007). A modified particle swarm optimization-based dynamic recurrent neural network for identifying and controlling nonlinear systems. Computers and Structures, 85(21-22), 1611-1622.

Goethals, I., Pelckmans, K., Suykens, J. A. K., & De Moor, B. (2005) Identification of MIMO Hammerstein models using least squares support vector machines. Automatica, 41, 1263–1272.

Hammar, K., Djamah, T., Bettayeb, M., (2015, Dec.). Fractional Hammerstein system identification using particle swarm optimization. 2015 7th International conference on modeling, identification and control (ICMIC) (pp. 1-6). Sousse, Tunisia: IEEE.

Han, M., Xi, J., Xu, S. & Yin, F. L. (2004) Prediction of chaotic time series based on the recurrent predictor neural network. IEEE Trans. on Signal Processing, 52(12), 3409-3416.

Hatalis, K., Alnajjab, B., Kishore, S., Lamadrid, A., (2014, Dec.). Adaptive particle swarm optimization learning in a time delayed recurrent neural network for multi-step prediction. 2014 IEEE symposium on foundations of Computational Intelligence (pp. 84-91). Orlando, FL, USA: IEEE.

Hou, Z., Shen, Q., & Li, H. (2003, DEC.). Nonlinear system identification based on ANFIS. IEEE international conference on neural networks and signal processing (pp. 510-512). Nanjing, China: IEEE.

Jiao, B., Lian, Z., & Gu, X. (2008). A dynamic inertia weight particle swarm optimization algorithm. Chaos, Solitions and Fractals, 37(3), 698-705.

Juang, C. F., & Chung, I. F. (2007). Recurrent fuzzy network design using hybrid evolutionary learning algorithms. Neurocomputing, 70(16-18), 3001-3010.

Kang, D., Lee, B., Won, S. (2007, Nov.). Nonlinear System Identification using ARX and SVM with Advanced PSO. The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON) (pp. 598-603), Taipei, Taiwan: IEEE.

Keesman, K. J. (2011). System Identification. New York, NY, Springer.

Kennedy, J., Eberhart, R. (1995, Nov.). Particle swarm optimization. IEEE International Conference on Neural Networks (pp. 1942-1948). Perth, WA, Australia: IEEE.

Khanesar, M. A., Shoorehdeli, M. A., Teshnehlab, M. (2007, Aug.). Hybrid training of recurrent fuzzy neural network model. IEEE International Conference on Mechatronics and Automation (pp. 2598-2603). Harbin, China: IEEE.

Ko, C. N., Lee, C. I., (2016, July). Identification of nonlinear system with outliers using modified quantum particle swarm optimization. 2016 International conference on system science and engineering (ICSSE) (pp. 1-4). Puli, Taiwan: IEEE.

Kristinsson, K. & Dumont, G. A. (1992) System identification and control using genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics, 22(5), 1033-1046.

Lee, C. H., & Teng, C. C. (2000). Identification and control of dynamic systems using recurrent fuzzy neural networks. IEEE Transactions on Fuzzy Systems, 8(4), 349-366.

Li, C., & Cheng, K. H. (2007). Recurrent neuro-fuzzy hybrid learning approach to accurate system modeling. Fuzzy sets and systems, 158, 194-212.

Lin, C. J., Chen, C. H. (2006). A compensation-based recurrent fuzzy neural network for dynamic system identification. European Journal of Operational Research, 172, 696-715.

Lin, C. T., & Lee, C. S. G. (1996). Neural fuzzy systems: a neuro-fuzzy synergism to intelligent system, Englewood Cliffs, NJ, Prentice-Hall.

Nickabadi, A., Ebadzadeh, M. M., & Safabakhsh, R. (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing, 11(4), 3658-3670.

Ojha, R., & Das, M. (2012). An adaptive approach for modifying inertia weight using particle swarm optimization. IJCSI International Journal of Computer Science Issues, 9(5), 105-111.

Shi, Y., & Eberhart, R. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. IEEE Evolutionary Computation, 1, 84-88.

Shi, Y., Eberhart, R. (1998, May). A modified particle swarm optimizer. IEEE International Conference on Evolutionary Computation proceeding (pp.69-73). Anchorage, AK, USA: IEEE.

Xu, X., Wang, F., Liu, G., Qian, F., (2013). Identification of Hammerstein systems using key-term separation principle, auxiliary model and improved particle swarm optimization algorithm. IET Signal Processing, 7(8), 766-773.

Yan, C. M., Guo, B. L., & Xu, X. X. (2012). Empirical study of the inertia weight particle swarm optimization with constraint factor. JSCSE International Journal of Soft Computing and Software Engineering, 2(2), 1-8.

Yang, X., Yuan, J., Yuan, J., & Mao, H. (2007). A modified particle swarm optimizer with dynamic adaptation. Applied Mathematics and Computation, 189(2), 1205-1213.

10 NOTES ON CONTRIBUTORS

Chung-Wen Hung received his B.S. degree from Feng Chia University, Taiwan, and his M.S. and Ph.D. degrees from the National Taiwan University, Taiwan. From 2009 to 2016, he was an assistant professor at National Yunlin University of Science and Technology, Taiwan. Since 2016, he has been an associate professor at the same university. His research interests are in motor control, internet of thing (IOT), signal processing, and MCU applications.

Wei–Lung Mao received a B.S. degree in electrical engineering from National Taiwan University of Science and Technology in 1994, and M.S. and Ph.D. degrees in electrical engineering from National Taiwan University in 1996 and 2004, respectively. He is now a professor in the Department of Electrical Engineering and Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology. His research interests are precision motion control, intelligent and adaptive control systems, satellite navigation systems, adaptive signal processing, neural networks, and communication electronics.

Han-Yi Huang received the B.S. degree in electrical engineering from Southern Taiwan University of Science and Technology in 2014, the M.S. degrees in electrical engineering from National Yunlin University of Science and Technology in 2017. His research interests are system identification, adaptive control system, fuzzy control system and precision control.