

A Nonlinear System Identification Method Based on Adaptive Neural Network

Junju Sun and Liyun Lin

The School of Mathematics and Computer Science, Xinyang Vocational and Technical College, Xinyang, China

Nonlinear system identification (NSI) is of great significance to modern scientific engineering and control engineering. Despite their identification ability, the existing analysis methods for nonlinear systems have several limitations. The neural network (NN) can overcome some of these limitations in NSI, but fail to achieve desirable accuracy or training speed. This paper puts forward an NSI method based on adaptive NN, with the aim to further improve the convergence speed and accuracy of NN-based NSI. Specifically, a generic model-based nonlinear system identifier was constructed, which integrates the error feedback and correction of predictive control with the generic model theory. Next, the radial basis function (RBF) NN was optimized by adaptive particle swarm optimization (PSO), and used to build an NSI model. The effectiveness and speed of our model were verified through experiments. The research results provide a reference for applying the adaptive PSO-optimized RBFNN in other fields.

ACM CCS (2012) Classification: Computing methodologies → Artificial intelligence → Knowledge representation and reasoning → Causal reasoning and diagnostics

Keywords: artificial neural network (ANN), nonlinear system identification (NSI), particle swarm optimization (PSO), generic model

1. Introduction

Since most of real-world physical systems are nonlinear in nature, it is of great significance for researchers of modern engineering and control engineering to evaluate and analyze nonlinear systems. As an important branch of nonlinear system research, nonlinear system identification (NSI) has made great research progress in recent years [1–6].

Despite their ability to describe system features, the iterative method, Hammerstein-Wiener model, and differential geometry cannot achieve desirable accuracy in system identification [7–9]. Artificial neural network (ANN) has attracted much attention at home and abroad, thanks to its powerful self-learning function and the ability to quickly find optimal solutions. With the aid of the ANN, it is possible to identify nonlinear systems accurately, and optimize the identification parameters [10–20].

With the development of intelligent control and optimization theory, various kinds of NSI methods have emerged based on neural networks (NNs) [21–23]. Kohler *et al.* [24] designed a three-layer deep convolutional neural network (D-CNN), and achieved good local optimal effects by identifying multivariable nonlinear systems with the D-CNN. Ang *et al.* [25] handled nonlinear systems with proportional-integral-derivative (PID) NN, and experimentally proved the excellence of PIDNN in approximating the objective function. Malhotra and Khanna [26] constructed a wavelet neural network (WNN), which adjusts the network weight adaptively online through least squares (LS) method. Based on multi-particle swarm genetic neural network, Poempool *et al.* [27] proposed an NSI method that enhances the learning ability of the model by increasing the number of hidden layer nodes.

The traditional ANN can effectively solve some NSI problems, but fail to realize ideal accuracy or training speed [28, 29]. Inspired by the foraging behavior of bird flocks, the particle swarm

optimization (PSO) has been developing rapidly in recent years, owing to its advantages in complex NSI (*e.g.* global search ability, simple implementation, and good performance). Phuchan *et al.* [30] introduced the PSO algorithm to optimize the design of backpropagation neural network (BPNN), and managed to improve the training and convergence speeds of the BPNN model. Based on the PSO, Adsawinnawanawa *et al.* [31] carried out linear regression in the weight adjustment of multilayer feedforward network, and realized the objective of solving the global optimal weights online.

This paper puts forward an NSI method based on adaptive NN, aiming to further improve the convergence speed and accuracy of the NN in NSI. Firstly, the basic ideas of NN-based NSI were sorted out. Then, the error feedback and correction of predictive control was combined with the generic model theory to create a nonlinear system identifier based on generic model. After that, an NSI model was designed by optimizing the radial basis function (RBF) NN through PSO, and the identification workflow was illustrated in a block diagram. Finally, the proposed model was proved effective and swift through experiments.

2. Principle of NN-Based NSI

The NN-based NSI is implemented by the following principle: First, a suitable NN model was selected as the forward model \hat{S} or inverse model \hat{S}^{-1} of the target nonlinear system S . Then, the weights of the connection matrix are continuously adjusted, making the error function approximating the minimum at a random precision. With self-learning and adaptation abilities, the NN was adopted to fit the dynamic or static system features, which are reflected in the input and output data of the system. In this way, the change trend of the identified system can be tracked adaptively.

The NN-based NSI needs to go through three stages: determining model structure, predicting the unknown parameters of the model, and verifying the model. The model structure is determined by selecting the number of network layers, identifying the number of nodes in each layer, and choosing suitable transfer functions and connection forms for the nodes. The mod-

el parameters are predicted to minimize the error function, while continuously adjusting the NN parameters. Here, the mean squared error (MSE) function is taken as the error function. In addition, model verification mainly verifies the generalization ability of the NN.

By the nature of identification, the NN-based NSI falls into two categories, namely, online identification and offline identification. In online identification, learning and training take place concurrently with the identification process; in offline identification, these operations are completed prior to identification. Online identification is characterized by the real-time-ness of the identification process. By contrast, the input and output of offline identification cannot cover the working range of the nonlinear system; any parameter change will cause a large error in identification, and even produce an erroneous model. To solve the problem, the two identification strategies need to be combined. That is, the offline identification should be implemented during online identification to acquire the weights of the connection matrix of the NN, and to make the acquired weights as the initial weights. This hybrid approach can greatly improve the online learning rate.

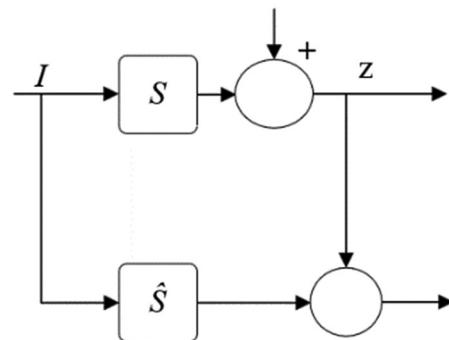


Figure 1. The model of the parallel structure.

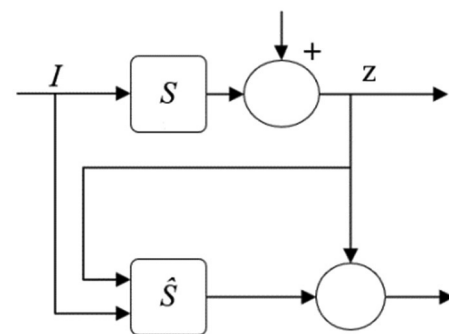


Figure 2. The model of the series-parallel structure.

By the type of feedback signal in the identification process, the NN-based NSI can be divided into the parallel structure in Figure 1 and the series-parallel structure in Figure 2. In the latter structure, the NN is trained with the input and output signals of the nonlinear system as the identification bases. The network training enhances the stability and convergence of the model. Therefore, the latter structure has been applied more widely than the former structure.

3. Construction of Generic Model for Nonlinear System

To solve the overdependence of NSI on adaptive control strategy, the error feedback and correction in predictive control were combined with the generic model theory to realize the centralized design of nonlinear modeling and control with error correction. The single-input-single-output nonlinear system with a time delay of 1 can be expressed as:

$$O(t) = f[O(t-1), \dots, O(t-n), I(t-1), \dots, I(t-m)] \quad (1)$$

where $I(t-1)$ and $O(t)$ are a set of observation data inputted to and outputted by system S at time t , respectively; $f(\cdot)$ is the functional representation of the nonlinear system.

Let \bar{I} be the set of all possible values of the set of observation data inputted to system S , and \bar{O} be the set of all possible values of the set of observation data outputted by system S based on the input dataset $I(t-1)$. Then, \bar{I} and \bar{O} can be viewed as the change domains of input and output of system S , respectively. Assuming that $\bar{I} \neq \bar{O}$, the characteristic parameter function $\phi(t)$, the pseudo gradient vector of output $O(t)$ with respect to input $I(t-1)$, must satisfy:

$$\phi(t) = \begin{cases} \frac{O(t) - O(t-1)}{I(t-1) - I(t-2)}, & u(t-1) \neq u(t-2) \\ \text{Constant } c, & u(t-1) = u(t-2) \end{cases} \quad (2)$$

For most nonlinear systems, the input and output obey the following equivalent transform:

$$O(t) = O(t-1) + \phi(t-1)[I(t-1) - I(t-2)]. \quad (3)$$

Therefore, it can be proved that, for the bounded ideal output signal $O'(t)$, the input signal $I(t)$

represented by formula (1) is controllable. Under the input signal $I(t)$, the nonlinear system will output signal $O(t)$. If t approaches ∞ , then $O(t)$ will infinitely approximate $O'(t)$. Suppose μ is a constant. Then, the nonlinear system (1) must satisfy the following condition at any time t :

$$|O(t) - O(t-1)| \leq \mu |I(t) - I(t-1)|. \quad (4)$$

If $I(t) \neq I(t-1)$, there must exist a time-varying characteristic parameter function $\phi(t)$ that satisfies $|\phi(k)| < \mu$. Then, the nonlinear system (1) can be linearized as:

$$O(t) = O(t-1) + \phi(t-1)[I(t-1) - I(t-2)] + \lambda b(t-1), \quad (5)$$

where $b(t-1)$ is the output error $O'(t-1) - O(t-1)$ of the model at time t ; λ is the error correction coefficient. Then, the time-varying pseudo-gradient vector $\phi(t)$ was regressed by k -order autoregressive (AR) model:

$$\hat{\phi}(t) = \rho_0 + \rho_1 \hat{\phi}(t-1) + \rho_2 \hat{\phi}(t-2) + \dots + \rho_k \hat{\phi}(t-k) + e(t) \quad (6)$$

where, ρ_0 is a constant term; ρ_1, \dots, ρ_k are the model parameters to be determined; $e(t)$ is the white noise. The predicted output of the system at time t can be expressed as:

$$\hat{O}(t) = \hat{O}(t-1) + \hat{\phi}(t-1)[I(t-1) - I(t-2)] + \lambda b(t-1). \quad (7)$$

As shown in formula (7), when the difference between $I(t-1)$ and $I(t-2)$, and the ideal output signal $O'(t)$ are known, the NSI can be converted into a regression problem, with the aim to rationalize the error correction coefficient λ and the pseudo gradient vector $\phi(t)$.

Figure 3 provides the structure of generic model-based nonlinear system identifier. To make the system output as shown in formula (7), the MSE function must be minimized as follows in the identification process:

$$MSE(t) = \sum_{t=1}^T [O'(t) - \hat{O}(t)]^2. \quad (8)$$

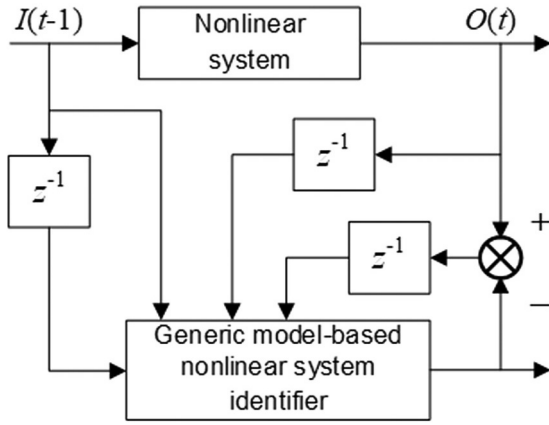


Figure 3. The structure of generic model-based nonlinear system identifier.

4. Adaptive NN Algorithm

4.1. RBFNN Identifier

The RBFNN boasts a simple structure and a fast convergence speed. Compared with other NNs, the RBFNN is powerful in function approximation, and excellent in identifying the dynamic features of nonlinear systems. Since the nonlinear system (1) can be identified with two structures (Figures 1 and 2), the RBFNN-based parallel and series-parallel models (Figures 4 and 5) can be respectively expressed as:

$$\hat{O}(t) = \hat{f} \left[\hat{O}(t-1), \dots, \hat{O}(t-n), \right. \\ \left. I(t-1), \dots, I(t-m) \right], \quad (9)$$

$$\hat{O}(t) = \hat{f} \left[O(t-1), \dots, O(t-n), \right. \\ \left. I(t-1), \dots, I(t-m) \right]. \quad (10)$$

As shown in Figures 4 and 5, the greatest difference between the RBFNN-based parallel and series-parallel models lies in whether the output signal enters the system as input signal. In the parallel model, the RBFNN is independent from the target system. Meanwhile, in the series-parallel model, the RBFNN operation is affected by the target system.

The basic structure of RBFNN consists of an input layer of n nodes, a hidden layer of p nodes, and an output layer of m nodes. Let $I = (I_1, I_2, \dots, I_n)$ be the input vector. Then, the

activation function of the j -th hidden node can be expressed as:

$$\varphi_j = \exp \left(-\frac{1}{2\sigma_j^2} \|I - c_j\|^2 \right) \quad (11)$$

where $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_p)$ is the variance matrix; $C = (c_1, c_2, \dots, c_p)$ is the matrix of cluster centers selected for k -means clustering (KMC). Let ω_{jk} be the connection weight between the j -th hidden layer node and the k -th output layer node. Then, the k -th predicted output of RBFNN can be expressed as:

$$O_k = \sum_{j=1}^p \omega_{jk} \exp \left(-\frac{1}{2\sigma_j^2} \|I - c_j\|^2 \right) \quad (12)$$

The parameters that RBFNN needs to solve include the center of the activation function, the variance, and the weights from the hidden layer to the output layer. Here, the center is selected through self-organized learning. Firstly, p cluster centers are selected for KMC. Suppose there are N training samples in which the vector of the l -th sample is I_l . As the vector is inputted to the network and judged whether it belongs to the j -th cluster center M_j , the j -th center in the matrix of cluster center C can be adjusted and updated by formula (13) after the t -th iteration:

$$c_j^r = \begin{cases} c_j^{r-1} + \gamma (I_l - c_j^{r-1}), & I_l \in M_j(I_l) \\ c_j^{r-1}, & I_l \notin M_j(I_l) \end{cases} \quad (13)$$

where $0 < \gamma < 1$ is the step length of the iterative process. Let c_{\max} be the maximum distance between the selected centers. Then, the j -th variance can be solved by:

$$\sigma_j = \frac{c_{\max}}{\sqrt{2p}} \quad (14)$$

The matrix of connection weights W was directly derived through the LS method, that is, the partial derivative of the loss function was solved and equated to zero. Suppose the network output vector equals the expected vector d . Then, the matrix of connection weights W can be simplified as:

$$W = \Phi^+ d \quad (15)$$

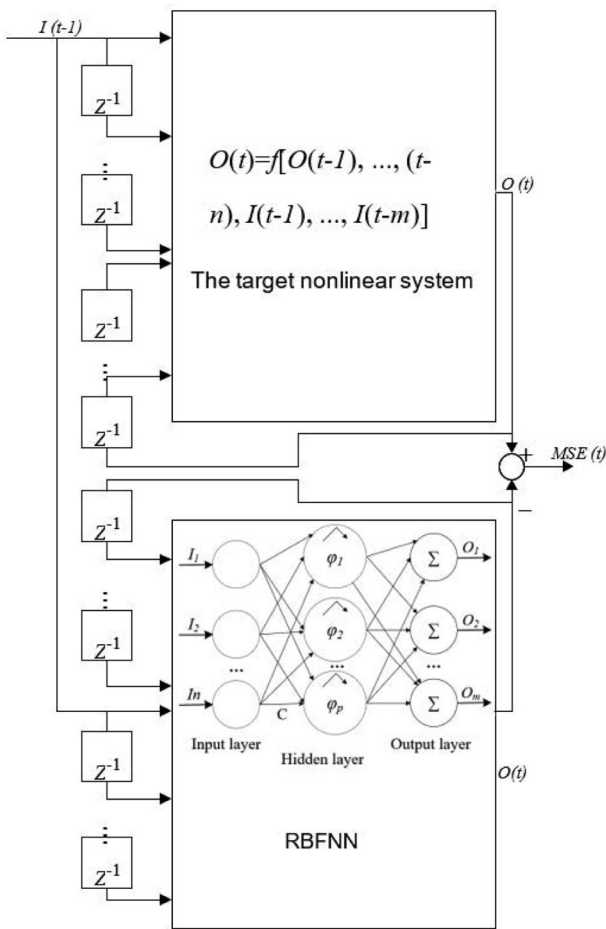


Figure 4. The RBFNN-based parallel structure.

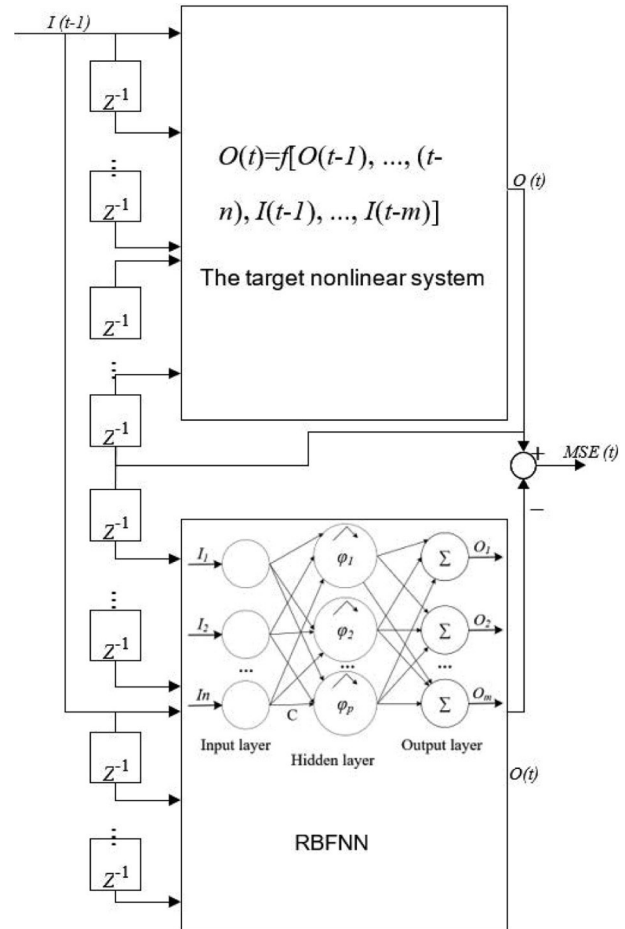


Figure 5. The RBFNN-based series-parallel structure.

where Φ^+ is the pseudo-inverse of matrix Φ , that is, the set $\{\varphi_j\}$ of output vectors of hidden layer nodes.

4.2. Adaptive PSO Optimization of Connection Weights

As part of the LS-optimization of matrix W , the calculation of the inverse matrix may affect the identification speed and accuracy. To speed up the convergence of the RBFNN, this paper chooses to optimize the connection weights with the adaptive PSO algorithm.

Suppose there is a particle swarm in a D -dimensional search space. The current positions of the M particles in the swarm can be expressed as $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$. The velocity of the i -th particle can be expressed as $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, where $i = 1, 2, \dots, M$. Moreover, the best-known positions of the i -th particles and the swarm can be respectively expressed as $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ and $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$.

Then, the i -th particle will iteratively update its velocity and position in the d -dimensional sub-space by:

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 [p_{id} - x_{id}(t)] + c_2 r_2 [p_{gd} - x_{id}(t)], \quad (16)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad (17)$$

where c_1 and c_2 are nonnegative acceleration factors; r_1 and r_2 are random numbers in $(0, 1)$; ω is the inertia weight.

To make particle search more pertinent, the v_{id} value was limited by the following criteria: If v_{id} is greater than the maximum value v_{\max} , make it equal to v_{\max} ; if v_{id} is smaller than the minimum value v_{\min} , make it equal to v_{\min} .

During the PSO-optimization of RBFNN, it is necessary to determine the dimensionality of the search space based on the total number of weights and thresholds of the network, and then establish the relationship between particle fitness and MSE:

$$fitness = \sqrt{\frac{1}{2N} \sum_{t=1}^T [O'(t) - \hat{O}(t)]^2} \quad (18)$$

To balance the local and global search abilities, the inertia weight should be relatively large in the early phase of optimization to ensure the global search ability, and relatively small in the later phase to accelerate the convergence. In addition, the inertia weights of particles with small fitness should be increased, and those of particles with large fitness should be decreased, making the particles with small fitness more active in the search for the optimal solution. To realize these goals, the inertia weight ω was adaptively adjusted by formula (19), such that the inertia weight gradually decreases in the optimization process and adaptively changes for particles with different fitness:

$$\omega = \begin{cases} \left(1 - \frac{t}{t_{\max}}\right) \omega_{\max} + \left(1 + \frac{t}{t_{\max}}\right) \omega_{\min}, & fitness \leq fitness_{avg} \\ \left(1 - \frac{t}{t_{\max}}\right) \omega_{\max} + \frac{t}{t_{\max}} \omega_{\min}, & fitness > fitness_{avg} \end{cases} \quad (19)$$

where t is the number of iterations; t_{\max} is the maximum number of iterations; ω_{\max} is the maximum inertia weight; ω_{\min} is the minimum inertia weight; $fitness_{avg}$ is the mean fitness of the swarm.

Furthermore, the acceleration factors must be adjusted to further speed up convergence and prevent the local minimum trap. In the early phase, the global search ability should be enhanced, using a large c_1 and a small c_2 . In the later phase, the convergence should be accelerated, using a small c_1 and a large c_2 . Therefore, the c_1 value was adaptively adjusted in a nonlinear manner by:

$$c_1 = \left(1 - \frac{t}{t_{\max}}\right) c_{\max} + \frac{t}{t_{\max}} c_{\min} \quad (20)$$

Meanwhile, the c_2 value was changed by the nonlinear relationship with the c_1 value:

$$c_2 = x^{-c_1} + y \quad (21)$$

where c_{\max} is the maximum acceleration factor; c_{\min} is the minimum acceleration factor; x and y are constants.

4.3. The Workflow of the NSI Based on Adaptive PSO-Optimized RBFNN

As shown in Figure 6, the NSI based on adaptive PSO-optimized RBFNN consists of seven steps:

Step 1: Initialize the model parameters, randomly set constant terms x and y , and determine the following parameters of the particle swarm: velocity range $[v_{\min}, v_{\max}]$; total number of particles M ; the maximum and minimum inertia weights ω_{\max} and ω_{\min} ; and the maximum and minimum acceleration factors c_{\max} and c_{\min} .

Step 2: According to the identification needs, construct the RBFNN model based on generic model, and determine the corresponding dimensionality D of the search space of the particle swarm.

Step 3: Establish the mapping from the initial position matrix X_i and the matrix of connection weights of the RBFNN W , and calculate the fitness of each particle.

Step 4: Update the best-known positions of each particle and the swarm P_i and P_g .

Step 5: According to the adjustment rules for adaptive inertia weights and acceleration factors, recalculate the ω , c_1 , and c_2 values of each particle, and, on this basis, compute and update the velocity v_{id} and position x_{id} of each particle.

Step 6: Judge if the termination condition is satisfied and map the global optimal position P_g of the swarm to the matrix of connection weights of the RBFNN W .

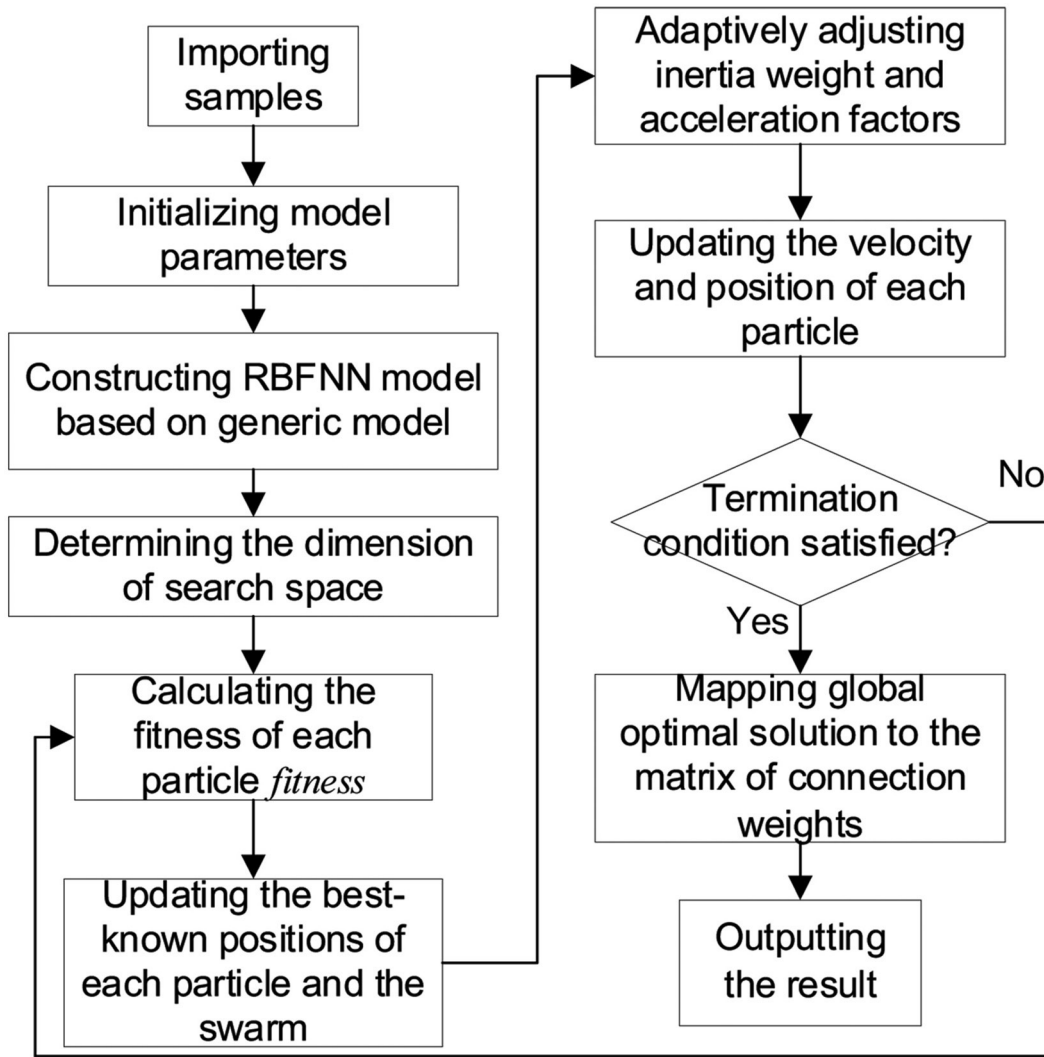


Figure 6. The workflow of the NSI based on adaptive PSO-optimized RBFNN.

Step 7: Input the test samples into the NN, analyze the output result, and verify the NSI performance of our model.

5. Experiments and Results Analysis

The experiments were carried out on a computer (CPU: Intel Core i8; memory: 16G; OS: Windows 10), using the simulation software Matlab 2013b. The structure of the target nonlinear system was illustrated by the generic model (5). The adaptive PSO algorithm was adopted to obtain the global optimal solution and the optimal values of unknown parameters through 200 iterations, before outputting the identification results of the NSI. Three different nonlinear

systems were identified, producing 1,400 data pairs for each system. The first 1,000 pairs were taken as the training set to train the NN, and the last 400 pairs were taken as the test set to verify the identification result. The differential equations of the three nonlinear systems are respectively as follows:

$$O(t) = 2 \sin(I(t-1)) + 5I(t-1) + 3I(t-1) \cdot \cos(I(t-1)), \quad (22)$$

$$O(t) = 2e^{I(t)} \cdot \sin(I(t-1)) + 5I(t-1), \quad (23)$$

$$O(t) = I(t-1) + \frac{O(t-1)[O(t-2)-3]}{7+O^2(t-1)}. \quad (24)$$

The identification results of our model and the traditional RBFNN, which is not optimized by adaptive PSO algorithm, on the first, second, and third nonlinear systems are displayed in Figures 7, 9, and 11, respectively. The MSEs of the identification of the first, second, and third NSIs are presented in Figures 8, 10, and 12, respectively.

As shown in Figures 8, 10, and 12, the MSEs exhibited a declining trend with the growing number of iterations. After 200 iterations, the MSE of the first nonlinear system dropped to and stabilized at 0.4525; the MSE of the second nonlinear system dropped to and stabilized at 0.3342. Therefore, the PSO-optimized RBFNN can control the number of iterations within 200 and achieve a fast convergence.

Comparing the identification performances of our model and the traditional RBFNN (Figures 7, 9, and 11), it can be seen that the output of our model deviated from the expected output by 3%, while that of the traditional RBFNN deviated from the expected output by 6%. Therefore,

our model achieved much better NSI effect than the traditional RBFNN.

6. Conclusion

This paper mainly designs an NSI method based on adaptive NN. After sorting out the basic ideas of NN-based NSI, a generic model-based nonlinear system identifier was created, which couples the error feedback and correction in predictive control with generic model theory. Next, the basic structures of the RBFNN for NSI were clarified, and the connection weights were optimized through adaptive PSO. After that, an NSI model was developed based on the adaptive PSO-optimized RBFNN, and the NSI workflow was detailed. Finally, the NSI results and MSEs of our model on three nonlinear systems were obtained through experiments. The results show that the proposed PSO-optimized RBFNN can control the number of iterations within 200, and achieve smaller identification error and better performance than the traditional RBFNN.

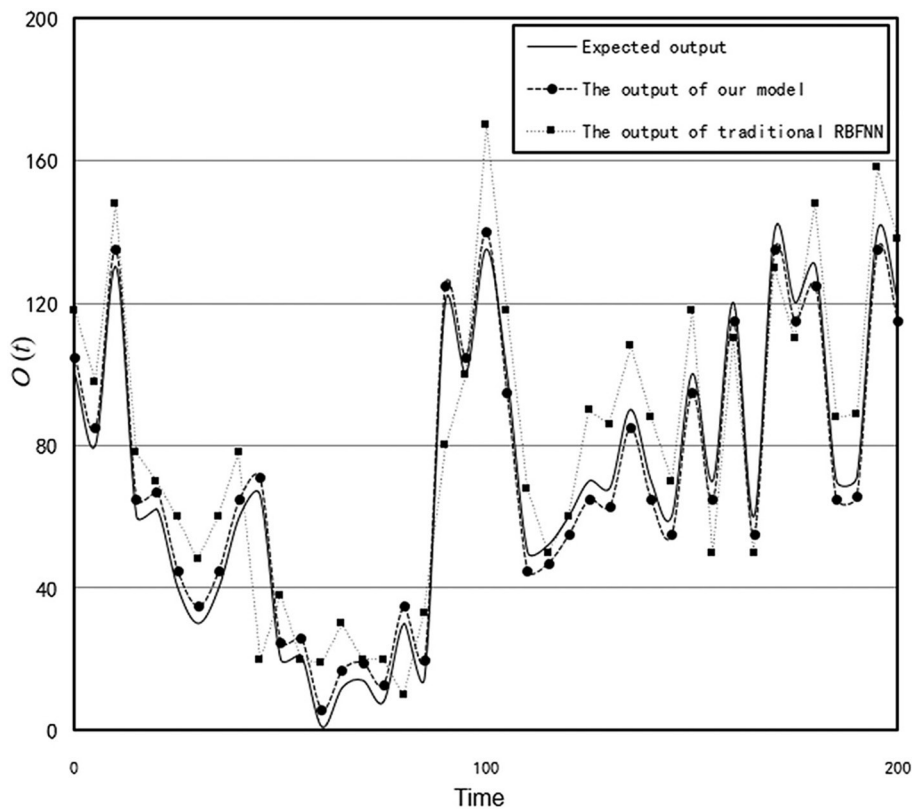


Figure 7. The identification results on the first nonlinear systems.

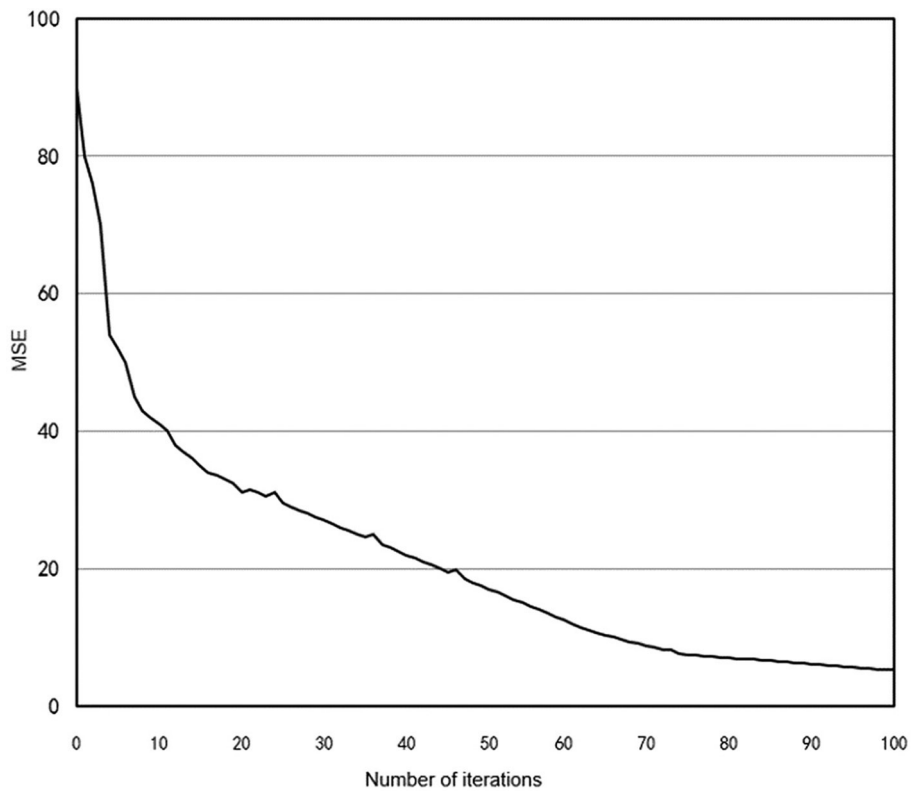


Figure 8. The MSE of the identification of the first nonlinear systems.

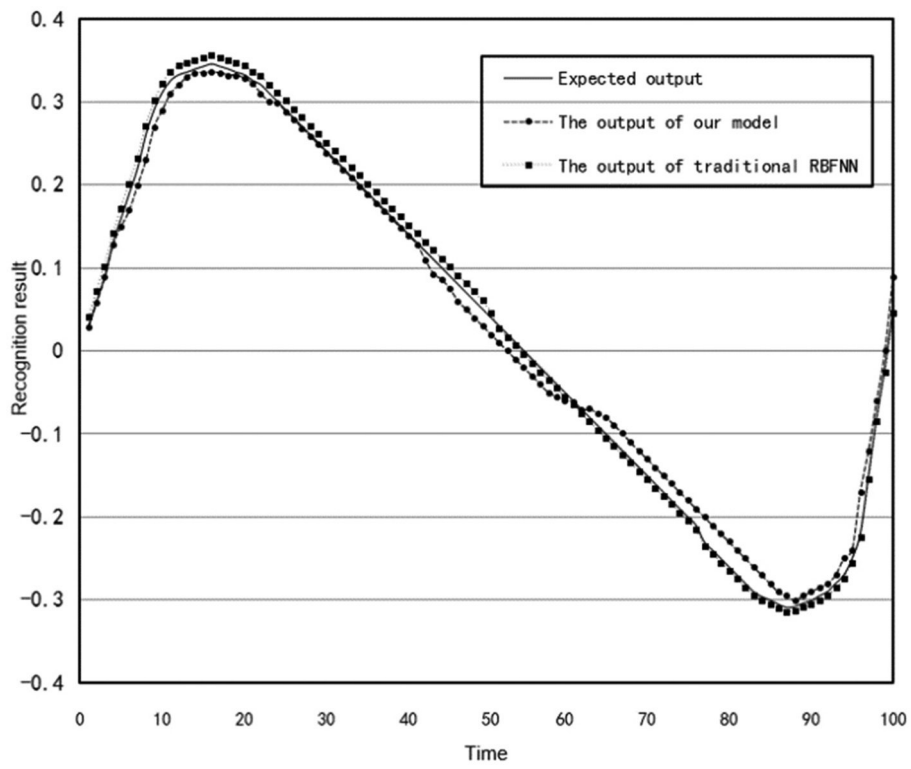


Figure 9. The identification results on the second nonlinear systems.

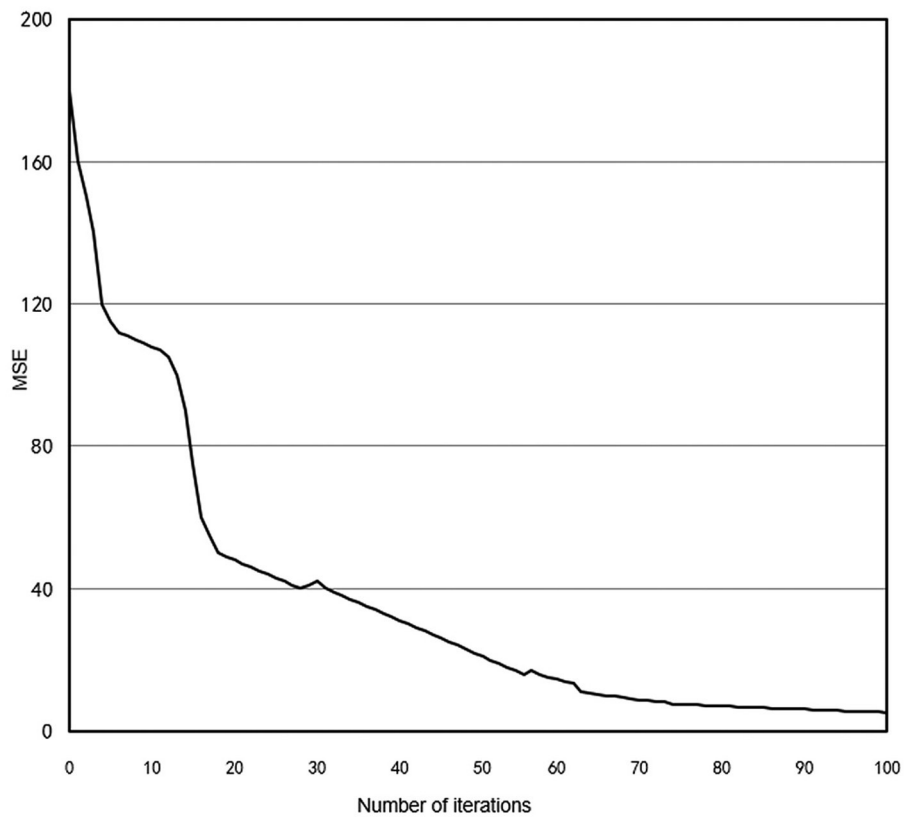


Figure 10. The MSE of the identification of the second nonlinear systems.

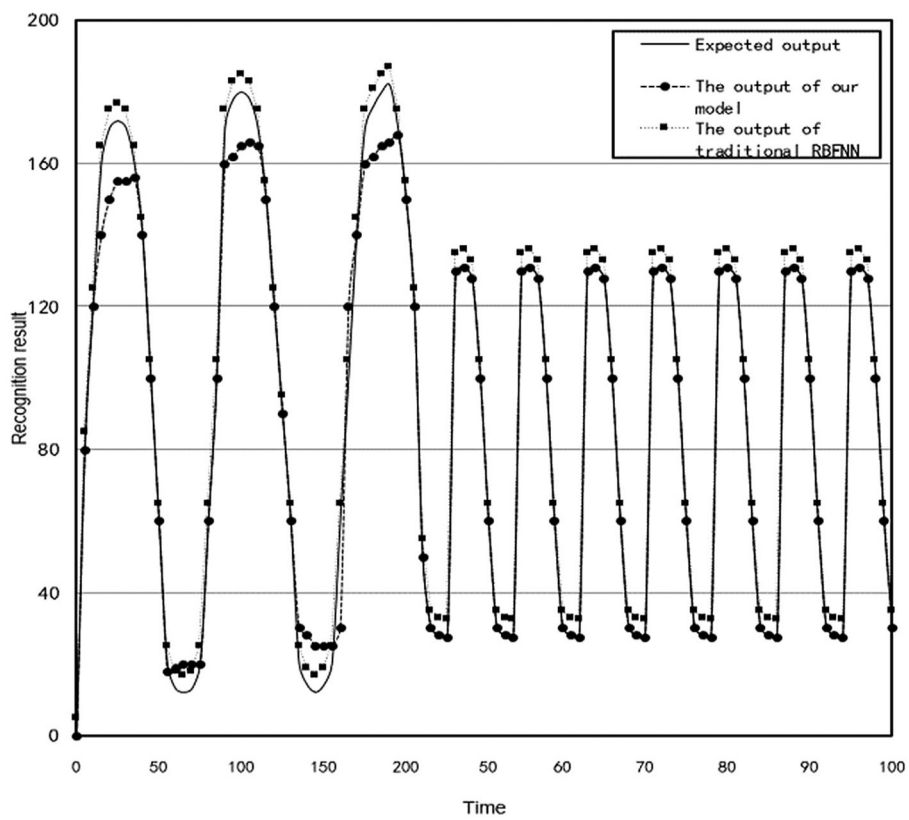


Figure 11. The identification results on the third nonlinear systems.

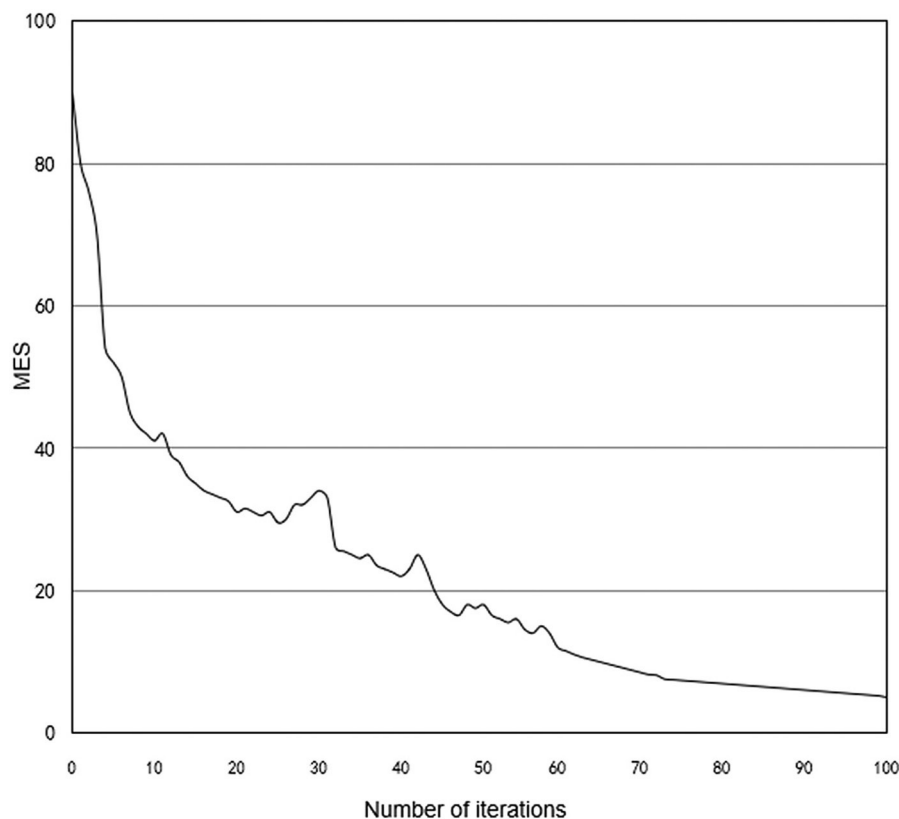


Figure 12. The MSE of the identification of the third nonlinear systems.

References

- [1] J. P. Noël and M. Schoukens, "Hysteretic Benchmark with a Dynamic Nonlinearity", *In Workshop on nonlinear system identification benchmarks*, pp. 7–14, 2016.
- [2] A. H. Ribeiro *et al.*, "On the Smoothness of Nonlinear System Identification", *Automatica*, vol. 121, pp. 109–158, 2020. <https://doi.org/10.1016/j.automatica.2020.109158>
- [3] M. Schoukens and J. P. Noël, "Three Benchmarks Addressing Open Challenges in Nonlinear System Identification", *IFAC-PapersOnLine*, vol. 50, no. 1, 446–451, 2017. <https://doi.org/10.1016/j.ifacol.2017.08.071>
- [4] F. Guo *et al.*, "Robust Identification of Nonlinear Errors-in-Variables Systems with Parameter Uncertainties Using Variational Bayesian Approach", *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3047–3057, 2017. <https://doi.org/10.1109/TII.2017.2712743>
- [5] Y. Sattar and S. Oymak, "Non-Asymptotic and Accurate Learning of Nonlinear Dynamical Systems", arXiv preprint arXiv: 2002.08538, 2020.
- [6] S. Singh *et al.*, "Learning Stabilizable Nonlinear Dynamics with Contraction-Based Regularization", arXiv preprint arXiv: 1907.13122, 2019.
- [7] F. Berkenkamp *et al.*, "Safe Learning of Regions of Attraction for Uncertain, Nonlinear Systems with Gaussian Processes", *In 2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 4661–4666. <https://doi.org/10.1109/CDC.2016.7798979>
- [8] I. R. Manchester and J. J. E. Slotine, "Control Contraction Metrics: Convex and Intrinsic Criteria for Nonlinear Feedback Design", *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053. <https://doi.org/10.1109/TAC.2017.2668380>
- [9] J. W. Jiang *et al.*, "Parameter Estimation for PMSM Based on a Back Propagation Neural Network Optimized by Chaotic Artificial Fish Swarm Algorithm", *International Journal of Computers Communications & Control*, vol. 14, no. 6, pp. 615–632, 2019. <https://doi.org/10.15837/ijccc.2019.6.3705>
- [10] S. Ouchtati *et al.*, "Novel Method for Brain Tumor Classification Based on the Use of Image Entropy

- and Seven Hu's Invariant Moments", *Traitement du Signal*, vol. 36, no. 6, pp. 483–491, 2019.
<https://doi.org/10.18280/ts.360602>
- [11] M. Schoukens and F. G. Scheiwe, "Modeling Nonlinear Systems Using a Volterra Feedback Model", In *Workshop on Nonlinear System Identification Benchmarks*, 2016.
- [12] J. B. Kim, "Implementation of Artificial Intelligence System and Traditional System: A Comparative Study", *Journal of System and Management Sciences*, vol. 9, no. 3, pp. 135–146, 2019.
- [13] F. Sbagoud *et al.*, "WPT-ANN and Belief Theory Based EEG/EMG Data Fusion for Movement Identification", *Traitement du Signal*, vol. 36, no. 5, pp. 383–391, 2019.
<https://doi.org/10.18280/ts.360502>
- [14] B. Mebarek and M. Keddou, "Prediction Model for Studying the Growth Kinetics of Fe₂B Boride Layers During Boronizing", *Ingénierie des Systèmes d'Information*, vol. 24, no. 2, pp. 201–205, 2019.
<https://doi.org/10.18280/isi.240212>
- [15] B. S. Kim and T. G. Kim, "Cooperation of Simulation and Data Model for Performance Analysis of Complex Systems", *International Journal of Simulation Modelling*, vol. 18, no. 4, pp. 608–619, 2019.
[https://doi.org/10.2507/IJSIMM18\(4\)491](https://doi.org/10.2507/IJSIMM18(4)491)
- [16] T. Mostefa *et al.*, "An Automatic Diagnosis Method for an Open Switch Fault in Unified Power Quality Conditioner Based on Artificial Neural Network", *Traitement du Signal*, vol. 35, no. 1, pp. 7–21, 2018.
<https://doi.org/10.3166/TS.35.7-21>
- [17] Z. Zhang *et al.*, "A Novel Job-Shop Scheduling Strategy Based on Particle Swarm Optimization and Neural Network", *International Journal of Simulation Modelling*, vol. 18, no. 4, pp. 699–707, 2019.
[https://doi.org/10.2507/IJSIMM18\(4\)CO18](https://doi.org/10.2507/IJSIMM18(4)CO18)
- [18] I. Choudira *et al.*, "Induction Machine Faults Detection and Localization by Neural Networks Methods", *Revue d'Intelligence Artificielle*, vol. 33, no. 6, pp. 427–434, 2019.
<https://doi.org/10.18280/ria.330604>
- [19] H. Choubey and A. Pandey, "Classification of Healthy, Inter-Ictal and Seizure Signal Using Various Classification Techniques", *Traitement du Signal*, vol. 35, no. 1, pp. 75–84, 2018.
<https://doi.org/10.3166/TS.35.75-84>
- [20] A. Svensson and T. B. Schön, "A Flexible State-Space Model for Learning Nonlinear Dynamical Systems", *Automatica*, vol. 80, pp. 189–199, 2017.
<https://doi.org/10.1016/j.automatica.2017.02.030>
- [21] S. N. Nguyen *et al.*, "A Neural Differential Evolution Identification Approach to Nonlinear Systems and Modelling of Shape Memory Alloy Actuator", *Asian Journal of Control*, vol. 20, no. 1, pp. 57–70, 2018.
<https://doi.org/10.1002/asjc.1529>
- [22] T. Deb *et al.*, "Design of Nonlinear Filters Using Affine Projection Algorithm Based Exact and Approximate Adaptive Exponential Functional Link Networks", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 11, pp. 2757–2761, 2019.
<https://doi.org/10.1109/TCSII.2019.2962916>
- [23] M. Kohler *et al.*, "PSO+: A New Particle Swarm Optimization Algorithm for Constrained Problems", *Applied Soft Computing*, vol. 85, pp. 105–865, 2019.
<https://doi.org/10.1016/j.asoc.2019.105865>
- [24] K. M. Ang *et al.*, "A Constrained Multi-Swarm Particle Swarm Optimization Without Velocity for Constrained Optimization Problems", *Expert Systems with Applications*, vol. 140, pp. 112–882, 2019.
<https://doi.org/10.1016/j.eswa.2019.112882>
- [25] R. Malhotra and M. Khanna, "Dynamic Selection of Fitness Function for Software Change Prediction Using Particle Swarm Optimization", *Information and Software Technology*, vol. 112, pp. 51–67, 2019.
<https://doi.org/10.1016/j.infsof.2019.04.007>
- [26] L. Poempool *et al.*, "Combine Multi Particle Swarm in Supporting Trapping in Local Optima", in *Proc. of the 2018 International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, 2018, pp. 1–4.
<https://doi.org/10.1109/ICEAST.2018.8434398>
- [27] H. Su, "Siting and Sizing of Distributed Generators Based on Improved Simulated Annealing Particle Swarm Optimization", *Environmental Science and Pollution Research*, vol. 26, no. 18, pp. 17927–17938, 2019.
<https://doi.org/10.1007/s11356-017-0823-3>
- [28] V. Cheypoca *et al.*, "The Use of Global Best Position in Rerun of Particle Swarm Optimization", in *Proc. of the 2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, 2018, pp. 1–4.
<https://doi.org/10.1109/ICEAST.2018.8434504>
- [29] W. Phuchan *et al.*, "Improved Rerun Particle Swarm Optimization Algorithm with Harmony Search", in *Proc. of the 2019 11th International Conference on Knowledge and Smart Technology (KST)*, 2019, pp. 46–50.
<https://doi.org/10.1109/KST.2019.8687810>
- [30] E. Adsawinnawanawa *et al.*, "Enhance Particle's Exploration of Particle Swarm Optimization with Individual Particle Mutation", in *Proc. of the 2019 7th International Electrical Engineering Congress (iEECON)*, 2019, pp. 1–4.
<https://doi.org/10.1109/iEECON45304.2019.8939047>

Received: January 2021

Revised: February 2021

Accepted: March 2021

Contact addresses:

Junju Sun
The School of Mathematics and Computer Science
Xinyang Vocational and Technical College
Xinyang
China
e-mail: sun_jj2021@126.com

Liyun Lin
The School of Mathematics and Computer Science
Xinyang Vocational and Technical College
Xinyang
China
e-mail: linliyun0410@163.com

JUNJU SUN received the BSc degree in computer science and technology from Henan University in 2004. In 2013, she was awarded the MSc degree in computer science and technology at Wuhan University of Technology. Since 2004, Sun has been a teacher at School of Mathematics and Computer Science, Xinyang Vocational and Technical College, currently serving as a lecturer. She was involved in the publication of three textbooks, and authored/co-authored more than 10 published papers on computer-related teaching and research. She is an expert in the application of computer-aided case-based teaching method, the application of digital media technology, and the neural network identification algorithms for nonlinear systems.

LIYUN LIN received the MSc degree from Shanghai Normal University in 2008. Two months later, she started working at Xinyang Vocational and Technical College. Currently, she holds the title of lecturer, and mainly engages in such fields as network and multimedia, and software technology.
