IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 3, 2020, accepted July 14, 2020, date of publication July 17, 2020, date of current version July 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009982

Identification of Dynamical Systems Using a Broad Neural Network and Particle Swarm Optimization

RAN HAN[1], **RONGJIE WANG**[1,2], **AND GUANGMIAO ZENG**[1]
[1]Marine Engineering Institute, Jimei University, Xiamen 361021, China
[2]Fujian Provincial Key Laboratory of Naval Architecture and Ocean Engineering, Xiamen 361021, China

Corresponding author: Rongjie Wang (roger811207@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51879118, in part by the Fujian Province Office of Science and Technology Support for Army under Grant B19101, in part by the Excellent Talent Support Program for Fujian Higher School in the New Century under Grant B17159, in part by the Scientific Research Foundation of Key Laboratory of Fishery Equipment and Engineering, Ministry of Agriculture of the People's Republic of China under Grant 2018001 and Grant 2016002, in part by the Scientific Research Foundation of Artificial Intelligence Key Laboratory of Sichuan Province under Grant 2017RYJ02, and in part by the Scientific Research Foundation of Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology under Grant 2017JSSPD01.

ABSTRACT System identification plays an important role in improving the structure and parameters of a system, but there are many problems encountered in actual operation. The identification of dynamic systems is not as simple as it is for static systems; thus, choosing effective model structures and parameters is the key to solving this problem. This paper proposes a novel algorithm based on a combination of a broad learning system (BLS) and particle swarm optimization (PSO) to identify nonlinear dynamical systems. The proposed method first uses the dimension expansion of the data set as the input of the BLS and then optimizes the model weight by the PSO algorithm. To verify the effectiveness of our proposal, we use four second-order systems for simulation experiments. The simulation results clearly show the efficiency and anti-interference ability of the proposed method.

INDEX TERMS Identification of dynamical system, broad learning system, particle swarm optimization.

I. INTRODUCTION

In the past several years, dynamic systems have been used in areas such as communication, control, and pattern recognition. The system identification method is used to establish the model of the controlled system, which can be used to analyze the performance and dynamic and static response characteristics of the system to improve the structure and parameters of the system; therefore, system identification has been widely considered by engineers, but they have also faced a number of problems in each of these application areas. Numerous engineering applications require an exact description of the dynamic behavior of the system under test. Dynamic models depicting the system of interest can be built utilizing the first principles of chemistry, physics, biology, and so forth. However, models developed in this way are difficult to derive, because they require detailed specialist knowledge, which may be lacking. In the area of control, the most common problem being encountered is dynamic system identification and control. Researchers from different fields have developed several methods to construct mathematical models for system identification. In system identification, observed input and output data is utilized to estimate dynamic models directly. Nonlinear dynamic behavior is presented by most of the real life systems, a linear model can not explain the dynamics of such systems for a large range of input and output values. So the first step in the process is to choose an efficient model. To address nonlinearity and various intelligent mathematical tools, such as those based on fuzzy logic or neural networks are quite popular. In [1], a mathematical tool based on fuzzy implications and reasoning is used to build a fuzzy model of a system. References [2]–[4] covers the most common and important approaches for the identification of nonlinear static and dynamic systems. Additionally, it provides the reader with the necessary background on optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Huiling Chen.

techniques making the book self-contained. The emphasis is put on modern methods based on neural networks and fuzzy systems without neglecting the classical approaches. The basic knowledge about the working principle of fuzzy systems is very well explained in [2], [3]. The second-order volterra model is used in [5], and it can effectively capture dynamic changes in the input-output data but does not perform well when identifying nonlinear systems. In [6], a neural network is used as an effective tool for the identification and control of nonlinear dynamical systems. Since most networks are affected by time-consuming training processes and complex structures, many studies require high performance and expensive equipment [7]–[10]. Recently, Chen and Liu developed a very fast and effective broad learning system (BLS) [11]. In the absence of a stacked-layer structure, the designed neural network broadly extends the neural nodes and incrementally updates the weight of the neural network when additional nodes are needed and as input data continuously enter the neural network [12]–[16]. Therefore, the BLS structure is well suited for modeling and learning in time-varying big-data environments [17], [18].

After determining the model of the system, the next step is to choose parameters. Although one can choose the least-squares method to estimate the parameters, if more accurate model parameters are needed, it is obvious that this method is not desirable. It is common to adjust the parameters through a dynamic back-propagation algorithm, but this method is slow to converge and possibly cannot converge. Reference [19] used a deep learning-based time-varying parameter identification model for composite load modeling with a ZIP load (It mainly consists of three parts. The first part is the constant impedance (Z) component. The second part is the constant current (I) component. The third part is the constant power (P) component and an induction motor. Reference [20] used diffusion systems with mode isolation parameters. This approach is shown to work for nonlinear reaction kinetics and on a variety of domains and surfaces. However, these two methods are not suitable for nonlinear system models. In [21], the force-displacement data are used to perform identification of the model parameters via a genetic algorithm (GA). The parameter identification method of the nonlinear dynamic system proposed in this paper is based on the PSO algorithm. The PSO algorithm is a powerful and widely used swarm intelligence technique, and it is easy to implement. Although the original PSO is very simple, with only a few parameters to adjust, it provides better performance in computing speed, computing accuracy, and memory size compared with other methods such as machine learning, neural network learning, and genetic computation. Hence, it has received much more attention in solving optimization problems [22]–[24]. As in [25], the PSO algorithm is used as an effective tool for parameter selection. In [26]–[28], the PSO algorithm was rapidly developed in various fields. In this paper, the model is built by using the BLS, and the PSO algorithm is used to obtain the parameters of the model. The purpose is to solve the problem that the nonlinear dynamic system is difficult to identify.

The main contributions of the proposed method can be summarized as follows. Firstly, we explore the application of new neural networks in the identification of nonlinear systems. this new neural network with simple structure, small number of parameters and fast update speed is undoubtedly better. Secondly, by introducing the PSO algorithm into the BLS, not only can the model update speed be accelerated, but also the model parameter accuracy can be improved.

The remainder of this paper is organized as follows. Section II provides the algorithm structure and implementation steps based on a PSO-BLS. Section III shows the simulation results. The final conclusions are drawn in Section IV.

II. PSO-BLS NONLINEAR DYNAMIC SYSTEM IDENTIFICATION MODEL

A. BROAD LEARNING SYSTEM

We present the input data $\mathbf{X}$ and project the data using $\Phi_j(\mathbf{X}W_{ei} + \beta_{ei})$ to represent the ith mapped feature $\mathbf{Z_i}$, where W_{ei} is the random weight with the proper dimensions. Similarly, the jth group of enhancement nodes, $\Phi_k(\mathbf{Z_i}\boldsymbol{W_{hj}} + \boldsymbol{\beta_{hj}})$, is denoted as $\mathbf{H_i}$. Φ_j and Φ_k can be different functions. The structure is illustrated in Figure. 1

In the BLS, W_{ei} can be adjusted by a sparse autoencoder (SAE). Thus, the ith mappings can be denoted as:

$$\mathbf{Z_i} = \phi_j(\mathbf{X}W_{ei} + \beta_{ei}), \quad i = 1, 2 \ldots n \tag{1}$$

The feature nodes are denoted as $\mathbf{Z^n} = [\mathbf{Z_1}, \ldots, \mathbf{Z_n}]$, where W_{hj} and β_{hj} are random weights. The enhanced nodes are denoted as:

$$\mathbf{H_i} = \phi_k(\mathbf{Z_i}W_{hi} + \beta_{hi}), \quad i = 1, 2 \ldots n \tag{2}$$

Therefore, the output of the BLS can be denoted as:

$$\begin{aligned}\mathbf{Y} &= [\mathbf{Z_1}, \ldots \mathbf{Z_n} \mid \mathbf{H_1}, \ldots, \mathbf{H_n}]\,\mathbf{W^n} \\ &= \left[\mathbf{Z}^n \,\middle|\, \mathbf{H}^n\right]\mathbf{W^n}\end{aligned} \tag{3}$$

$\mathbf{W^n}$ represents the output layer weight. For a single input system, the input $\mathbf{X}(k)$ $(k = 1, 2 \ldots n)$ can refer to (4), and the enhanced node can refer to (5). The BLS output is shown by (6).

$$\mathbf{X}(k) = [x_1, x_2 \cdots x_a] = [1, x(k) \cdots 2x(k)x_{a-1} - x_{a-2}] \tag{4}$$

where $a(a > 2)$ represents the dimension of the required extension. This is an artificially set constant. According to experimental experience, we generally set it to 4.

$$\mathbf{H} = \phi\left(\mathbf{W_1X} + \mathbf{b_1}\right) \tag{5}$$

$\mathbf{W_1}$ and $\mathbf{b_1}$ are random input weights. ϕ is activation function. $\mathbf{C}$ is enhanced node.

$$\mathbf{Y} = \mathbf{W_2} \cdot [\mathbf{X} \mid \mathbf{H}] \tag{6}$$

$\mathbf{W_2}$ is the output weight. The BLS Input consists of enhanced nodes C and input X**. S** is the output of the model.

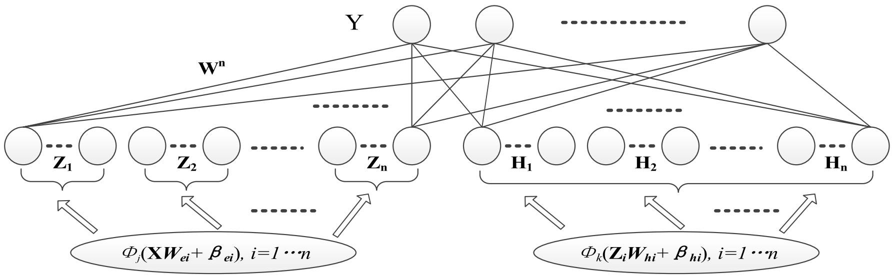

FIGURE 1. Structure of the BLS.

The principle is shown in Figure 2. As can be seen, if the input is a low-dimensional signal, we do not need to extract the feature vector of the signal as the BLS input.

The broad learning construction model and learning procedure are listed in **Algorithm 1**. To solve the problems of overfitting during network training and the slow convergence speed. In this paper, we used PSO to optimize the output weight $\mathbf{W_2}$ to improve the accuracy of the whole model for nonlinear system identification.

Algorithm 1 Broad Learning Model: Addition and Enhancement Nodes

Input : X
Output: Y
For $i = 1; i < a + 1$ **do**
    Random $\mathbf{W_1}$, **b1**;
    Calculate $\mathbf{H}_i = \phi(\mathbf{W_1 X}_i + \mathbf{b_1})$ with Eq. (5);
end
Set the enhancement mapping group $\mathbf{H}^k$;
$\mathbf{H}^k = [H_1, \ldots, H_a]^k$
Set $\mathbf{Z}^k = [\mathbf{X}^k, H^k]$ as Broad Learning model input
For $k = 1; k < n + 1$ **do**
    Random $\mathbf{W_2}$;
    Calculate $\mathbf{Y(k)} = \mathbf{W_2 Z}^k$ with Eq. (6);
    Calculate $\mathbf{W_2}$ by **Algorithm 2**
end

Algorithm 2 Particle Swarm Optimization (PSO): Using PSO to Find the BLS Weight Coefficient

Input : Speed; Iteration; Size
Output: W_2
While the training error threshold is not satisfied **do**
Find the best individual value (gbest) and the best group value
for $i = 1$: maxgen
    for $j = 1$: sizepop;
        Randomly generate a group (pop)
        Update speed and particle
        V = c_1 * rand(gbest-pop) + c_2 * rand(zbest-pop)
        Calculate the new particle fitness values
        if fitness(j) < fitnessgbest(j)
        gbest(j,:) = pop(j,:);
        fitnessgbest(j) = fitness(j);
        end
        if fitness(j) < fitnesszbest(j)
        zbest(j,:) = pop(j,:);
        fitnesszbest(j) = fitness(j);
        end
    end
end
Set **zbest** = $\mathbf{W_2}$;

B. DESIGNING STEPS FOR PSO-BLS

PSO is a biology-inspired evolutionary computation algorithm that was first introduced by Kenndy and Eberhart [31]. It is a population-based stochastic optimization technique inspired by the boid model. The boid model was introduced by Reynolds in 1987 and was inspired by the aggregate motion of a flock of birds [25]. The algorithm has the converges quickly, has an uncomplicated update mechanism, and is easy to program. The algorithm begins by randomly generating an initial population. This population is made up of many particles that are candidate solutions to the optimization problem. Some scholars have analyzed the convergence of the PSO [32]–[34], The latest research analyzed the convergence of the PSO by using the martingale theory (The martingale theory is one of the pillars of modern probability theory and a basic tool in applications) [32], and their results show that PSO can reach the global convergence with probability one. In our question, the task is to find the output weight matrix $\mathbf{W_2}$ required by the model. According to the size of the defined cost function, each particle optimal solution (gbest) and population optimal solution (zbest) is recorded for the particle velocity update.

The overall structure of the PSO-BLS nonlinear system identification method is shown in Figure 3. According to the characteristics of the BLS model, this paper improves the traditional PSO algorithm. As shown in **Algorithm 2**, the problem of slow convergence due to fixed parameters is compensated for by randomly generating parameters c_1 and c_2. $E(k)$ in the figure indicates that the profit and loss

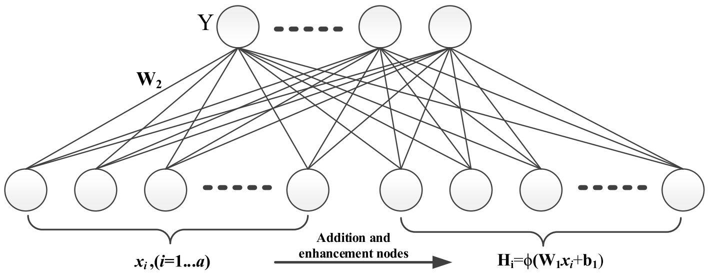

FIGURE 2. Structure of PSO-BLS.

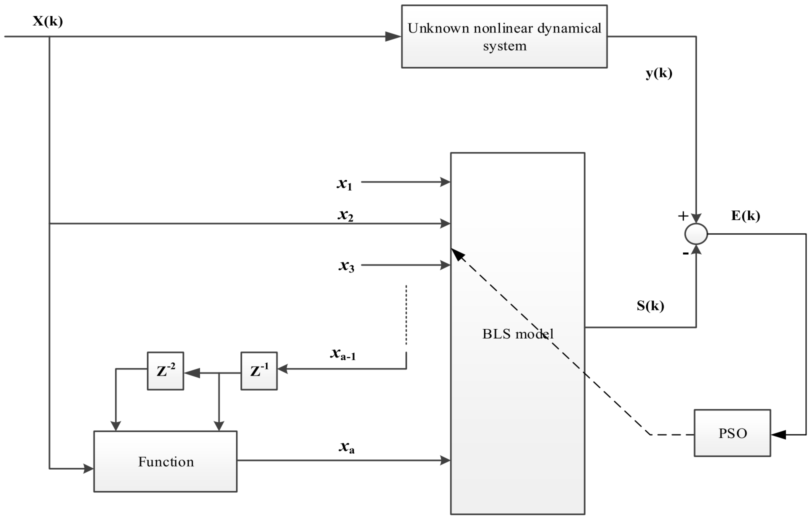

FIGURE 3. Overall structure of PSO-BLS modeling with a second-order system.

function of the model is shown by (7).

$$E=\frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{m}\left\|S_{k,j}-y_{k,j}\right\|^{2} \tag{7}$$

where m represents the output dimension. n represents the number of test samples, and y represents the expected output of the model. S represents the real output of the model. The function module is shown in **Algorithm 3**. The input expansion method is inspired by [35], we use $x(k)$ to represent the value of the kth data point in the manuscript which is known by the system. For example, we extend the first data point to a vector. Assuming $a = 4$, the first data point ($k = 1$) can be expanded to $\mathbf{X}(1) = [x_1, x_2, x_3, x_4]$, where $x_1 = 1$, $x_2 = x(1)$, $x_3 = 2x(1)x_2\text{-}x_1$, $x_4 = 2x(1)x_3\text{-}x_2$.

The design steps for PSO-BLS modeling are outlined as follows:

Step 1: Determine the output type of the system (single-input single-output, single-input multiple-output, multiple-input multiple-output). Determine the dimension of the input extension and process input data through **Algorithm 3**.

Step 2: Extend the input by the BLS (**Algorithm 1**) and calculate the model fitness function E. The relationship between the fitness function and the output weight $\mathbf{W_2}$ is determined.

Step 3: Use the PSO algorithm to optimize the output weight $\mathbf{W_2}$ to find the global optimal solution, as shown in **Algorithm 2**.

Step 4: After determining the model parameters of the PSO-BLS system, we will convert the test data through the function module as system's input and predict the output of the system.

III. VERIFICATION TESTS

A. RESULTS OF THE PROPOSED METHOD

In the following, the parameter settings for the simulation experiments are shown in Table 1. The maximum number of iterations (maxgen) of PSO and differential evolution (DE) is set to 500, the initialization parameters (c_1, c_2) are set to random numbers between 0 and 1, and the population size of the particle swarm is 100. The GA cross probability (CP) is set to 0.4, the variation probability (VP) is set to 0.2, and the selection stage coefficient (S) is set to 10. To prove that the PSO-BLS model has better test accuracy and less convergence error. We choice four types of second-order systems which have obvious nonlinear characteristics. We set

Algorithm 3 Function

```
Input : training data X(k), k = 1, 2, 3 ... n;
Output: x_i
Set the group X^k = [x_1, x_2,...x_a]^k
for i = 1: a
    if i == 1;
        x_1 = 1;
    end
    if i == 2
        x_2 = x(k);
    end
    if i > 2
        x_i = 2x(k) x_{i-1} - x_{i-2}
    end
end
Set X = x_i; X as the PSO-BLS model input
```

TABLE 1. Experimental parameters.

Parameter	c_1 rand(0,1)	c_2 rand(0,1)	maxgen 500
	sizepop 100	a 4	n 400/700
	CP 0.4	VP 0.2	S 10

the input of the model in Examples 1-3 as:

$$x(k) = \begin{cases} \sin\ 2\pi k/250 & \text{for } k \le 250 \\ 0.8\sin\ 2\pi k/250 + 0.2\sin\ 2\pi k/25 & \text{for } k > 250 \end{cases} \tag{8}$$

Example 1: The second-order system is described by a difference equation as:

$$\begin{cases} y(k+1) = 0.3y(k) + 0.6y(k-1) + g[x(k)] \\ g(\cdot) = 0.6\sin(\pi x(k)) + 0.3\sin(3\pi x(k)) \\ \qquad + 0.1\sin(5\pi x(k)) \end{cases} \tag{9}$$

Example 2: The second-order system is described by a difference equation as:

$$\begin{cases} y(k+1) = f[y(k), y(k-1)] + x(k) \\ f(a,b) = ab(a+0.5)(b-1)/(1+a^2+b^2) \end{cases} \tag{10}$$

Example 3: The second-order system is described by a difference equation as:

$$\begin{cases} y(k+1) = f(y(k)) + g(x(k)) \\ f(y(k)) = y(k)[y(k)+0.3]/[1+y(k)^2] \\ g(x(k)) = x(k)[x(k)+0.8][x(k)-0.5] \end{cases} \tag{11}$$

Example 4: The second-order system is described as:

$$x(k) = \begin{cases} \sin(2\pi k/25) & k \le 250 \\ 1 & k \le 500 \\ -1 & k \le 750 \\ 0.3\sin(k\pi/25) + 0.1\sin(k\pi/32) & \\ \qquad +0.6\sin(k\pi/10) & k \le 1000 \end{cases} \tag{12}$$

$$\begin{cases} y(k+1) = f(y(k), y(k-1), y(k-2), \\ x(k), x(k-1)) \\ f(a,b,c,d,e) = abce([(c-1)+0.5] \\ +x(k+1))/(1+b^2+c^2) \end{cases} \tag{13}$$

For example, we can obtain the input and output of the second-order system form example 1-4. In example1, we select the first 400 data points as training samples. Firstly, we expand the original data points into a vector $\mathbf{X}$ ($a = 4$, $\mathbf{X}(k) = [\ x_1, x_2, x_3, x_4]$). Secondly, we obtain the enhanced node $\mathbf{H}$ through equation (5), using [$\mathbf{X}$, $\mathbf{H}$] as the input of the PSO-BLS model (the input dimension is $2a$), and the model output is the output of the known second-order system. Then the BLS weights are optimized by the PSO algorithm. Finally, we use PSO-BLS network model to predict the output of 200 training samples. We give the visual comparison on four cases in Fig. 4-26, the simulation results clearly show that in terms of prediction accuracy. The PSO-BLS model used in this paper has smaller convergence errors and faster convergence speed. The detailed experimental data are shown in Table 2. As we can see, the number of test samples is 200 or 300, we predict the output of these test samples at the same time. PSO-X: Neural network weights are optimized by PSO. DE-BLS: BLS model weights are optimized by DE. GA-BLS: BLS model weights are optimized by the GA. DE-X: Neural network weights are optimized by DE. GA-X: Neural network weights are optimized by GA. Based on the above four groups of experiments, this article adopts the PSO-BLS method as the object of the key test. During the recognition process, by expanding the original input signal, it is found that the input dimension is not high, so there are fewer network weight parameters to be trained. Therefore, the use of optimization algorithms to optimize the weight parameters may better accord with our requirements, and the results of simulation experiments also verify this. In terms of prediction accuracy, the PSO-BLS model used in this paper has smaller convergence errors and faster convergence speed.

The performance of DE-BLS and GA-BLS is poor. The main reason is that DE and GA have instability in dealing with nonlinear problems. The DE algorithm is essentially a multiobjective (continuous-variable) optimization algorithm used to solve the overall optimal solution in a multidimensional space. During the variation process, individuals are randomly selected from the population, resulting in the partial loss of information, and the BLS structure leads to an increase in input information; then, the instability is more obvious. GA and DE face the same situation. The results of the PSO-BLS experiment are better because

TABLE 2. Comparison results obtained from the proposed method and the other methods.

Plant/Model	PSO-BLS	PSO-X	DE-BLS	DE-X	GA-X	GA-BLS	BLS	No. of samples	
								Training	Testing
Example 1	0.1003	0.1043	0.1682	0.2157	2.7974	2.9361	0.1267	400	200
Example 2	0.0264	0.0543	0.12	0.05	0.1645	0.2539	0.1511	400	200
Example 3	0.1038	0.1050	0.1486	0.1113	0.1203	0.1382	0.2264	400	200
Example 4	0.0219	0.1409	0.2835	0.122	0.1475	0.2807	0.09	700	300
Iteration	k<80	k>100	k>100	k>100	k>100	k>100	k>100	/	
Input variables	8	4	8	4	4	8	8		

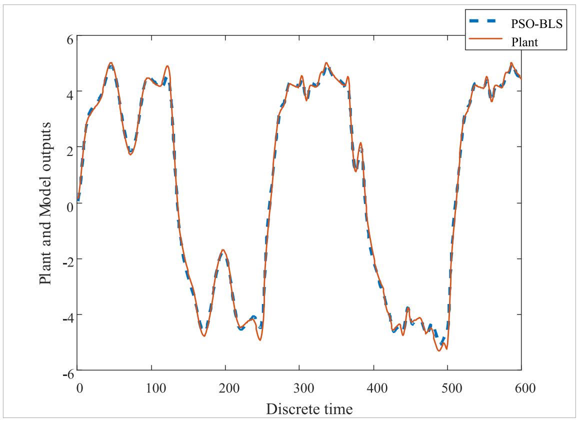

FIGURE 4. Modeling result of Example 1 based on PSO-BLS.

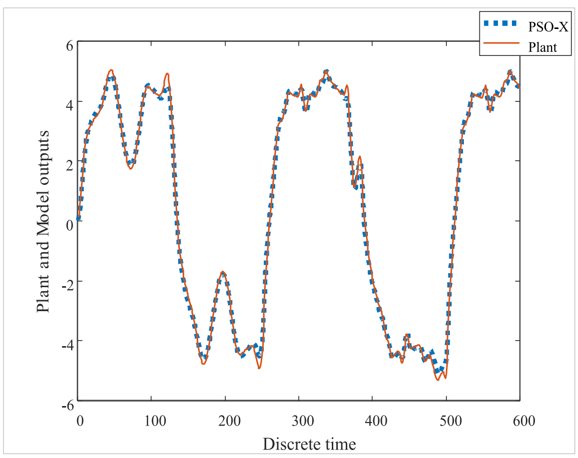

FIGURE 5. Modeling result of Example 1 based on PSO-X.

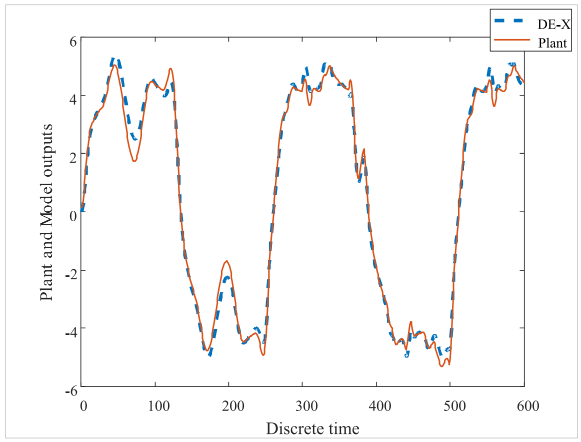

FIGURE 6. Modeling result of Example 1 based on DE-X.

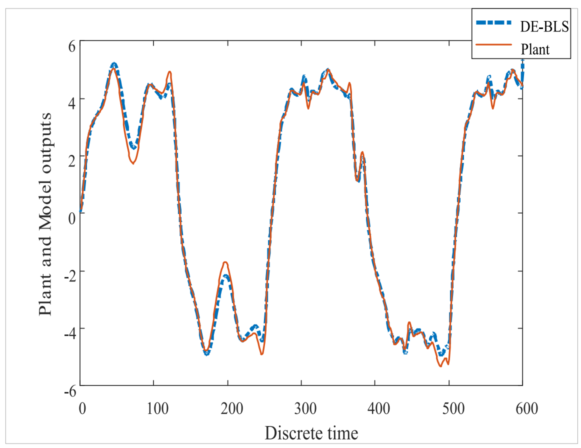

FIGURE 7. Modeling result of Example 1 based on DE-BLS.

PSO is essentially a group search optimization algorithm, which is more suitable for dealing with continuous or discrete space optimization problems. The BLS structure increases the amount of input information, highlighting the advantages of the PSO algorithm in dealing with nonlinear problems.

B. COMPARISON WITH OTHER METHODS

To verify the anti-interference ability of the model, predictive experiments were carried out by adding white gaussian noise to the input samples with signal-to-noise ratios of 35 dB, 30 dB, 25 dB, 20 dB, and 15 dB. The error calculation method is the same as the previous section, the specific

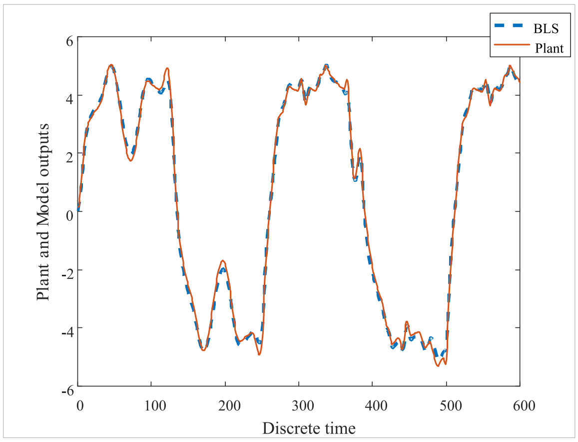

FIGURE 8. Modeling result of Example 1 based on the BLS.

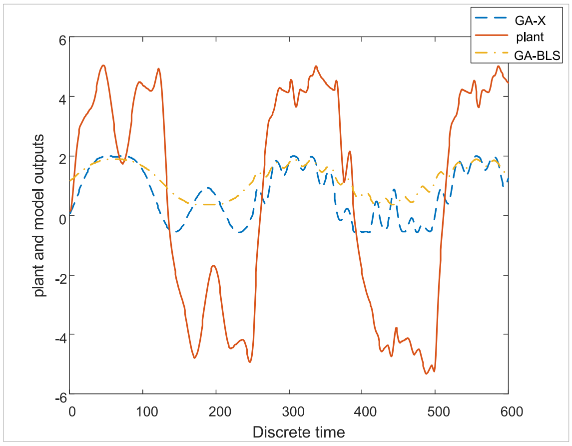

FIGURE 9. Modeling result of Example 1 based on GA-BLS and GA-X.

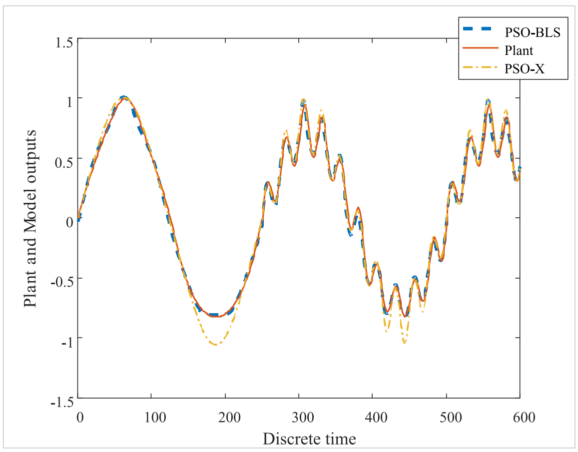

FIGURE 10. Modeling result of Example 2 based on PSO-BLS and PSO-X.

detail can refer to (7). As shown in Figure 26, two different optimization methods are used in [29] and [30]: differential evolution and PSO. To ensure the accuracy of the experiment, the training model and parameter setting method used in the paper are adopted in the comparison experiment. The biggest differences from [30] are the extension function for the input in the paper and the model structure.

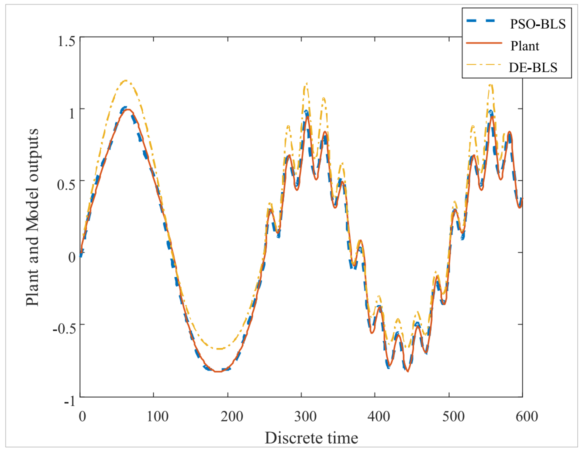

FIGURE 11. Modeling result of Example 2 based on PSO-BLS and DE-BLS.

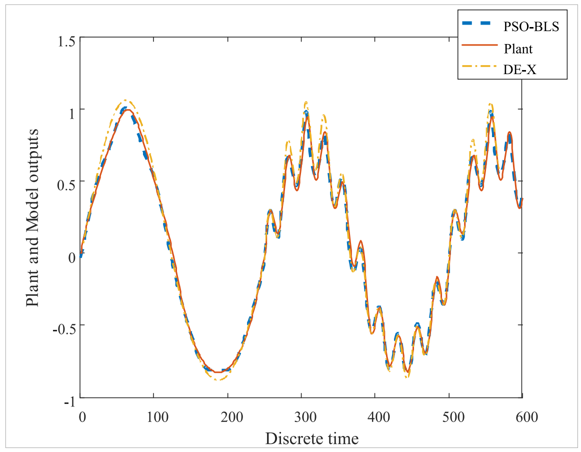

FIGURE 12. Modeling result of Example 2 based on PSO-BLS and DE-X.

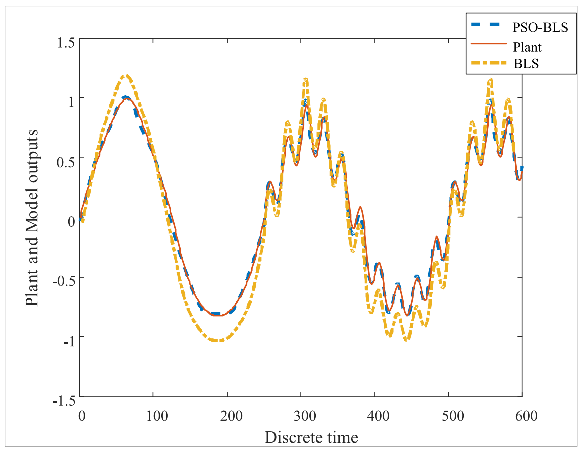

FIGURE 13. Modeling result of Example 2 based on PSO-BLS and BLS.

Reference [30] uses the most common three-layer network model, but the choice of hidden layer dimensions is different from that of the traditional network model. In [21], the force-displacement data are used to perform the identification of the model parameters via GA. Comparing the proposed method with the other three methods, the same

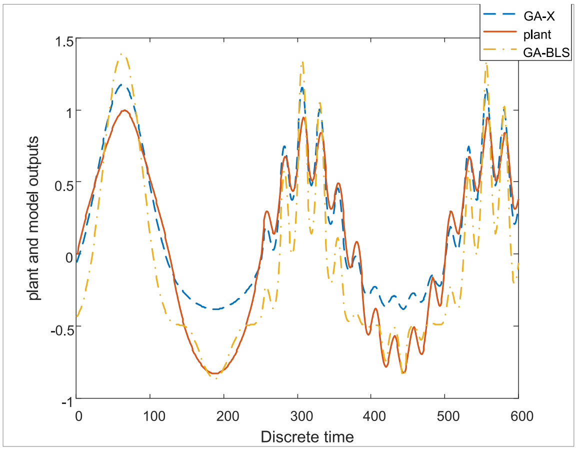

FIGURE 14. Modeling result of Example 2 based on GA-BLS and GA-X.

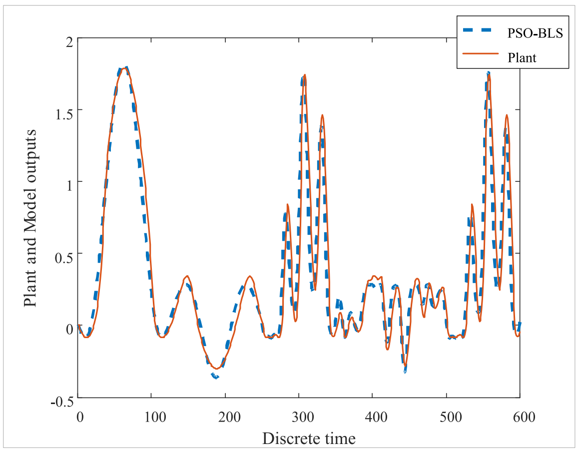

FIGURE 15. Modeling result of Example 3 based on PSO-BLS.

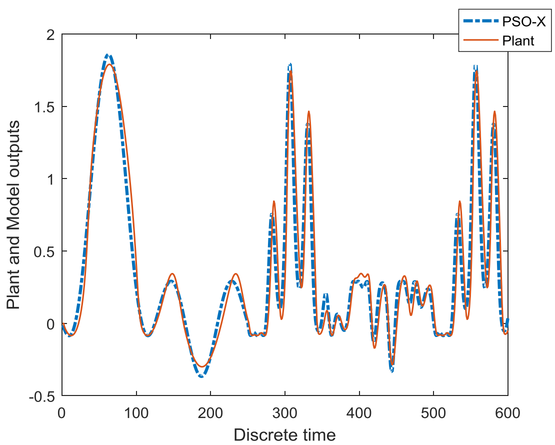

FIGURE 16. Modeling result of Example 3 based on PSO-X.

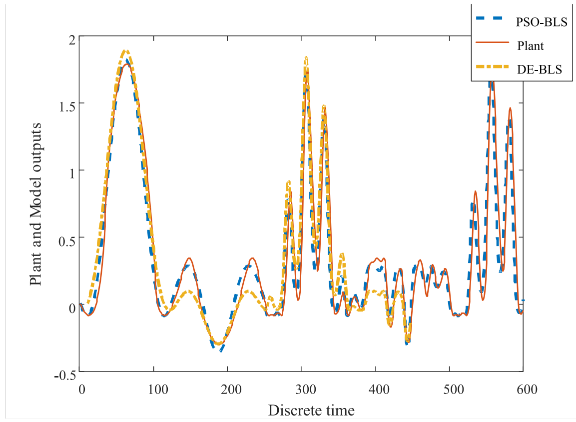

FIGURE 17. Modeling result of Example 3 based on PSO-BLS and DE-BLS.

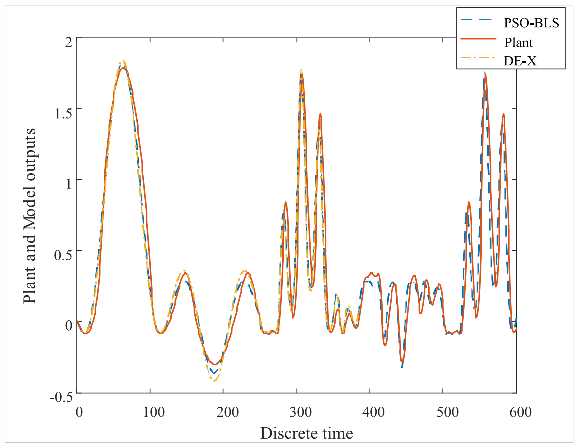

FIGURE 18. Modeling result of Example 3 based on PSO-BLS and DE-X.

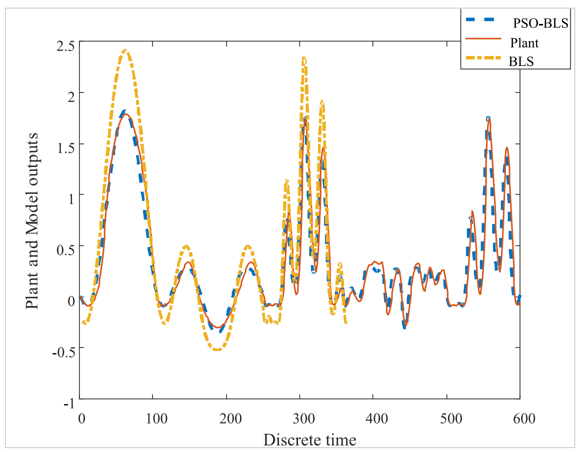

FIGURE 19. Modeling result of Example 3 based on PSO-BLS and BLS.

four second-order systems mentioned in the previous section are used. To ensure the validity of the results, each task was repeated 5 times, and the corresponding results are shown in Figure 26. We add noise to the input to simulate the interference of the external environment on the signal. When we extract the sample signal, there may be an error with the actual signal. In the training process of the model, we use the actual signal without noise. In the prediction process, we will add noise to the signal to observe whether the output of the

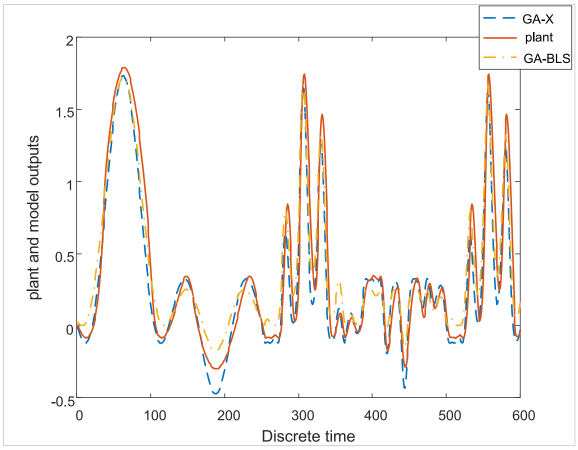

FIGURE 20. Modeling result of Example 3 based on GA-BLS and GA-X.

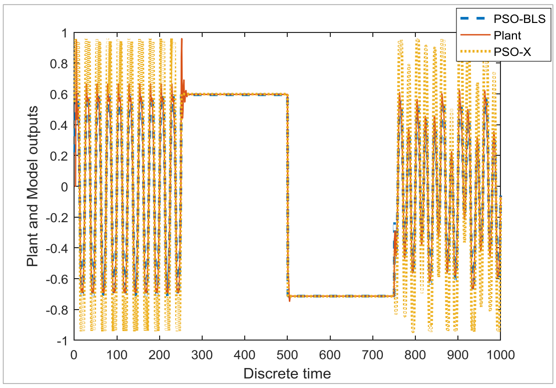

FIGURE 21. Modeling result of Example 4 based on PSO-BLS and PSO-X.

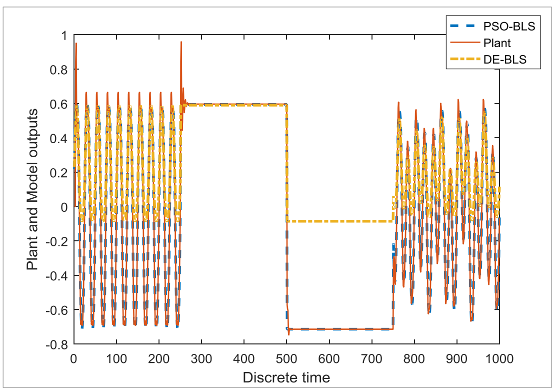

FIGURE 22. Modeling result of Example 4 based on PSO-BLS and DE-BLS.

model will be affected. Simulation experiments show that the PSO-BLS model is less affected by interference.

The traditional network structure does not adapt to current needs, mainly because of changes in input information. The input information of a nonlinear system is artificially extended. If the input information dimension is not large, the traditional mapping layer will compress the input information, which is not conducive to building a system model.

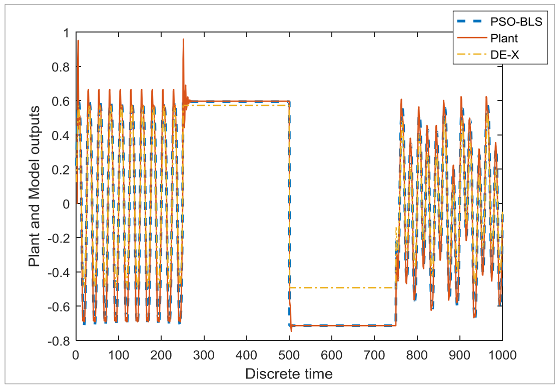

FIGURE 23. Modeling result of Example 4 based on PSO-BLS and DE-X.

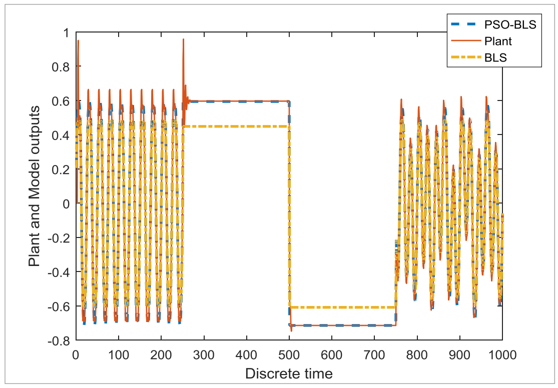

FIGURE 24. Modeling result of Example 4 based on PSO-BLS and BLS.

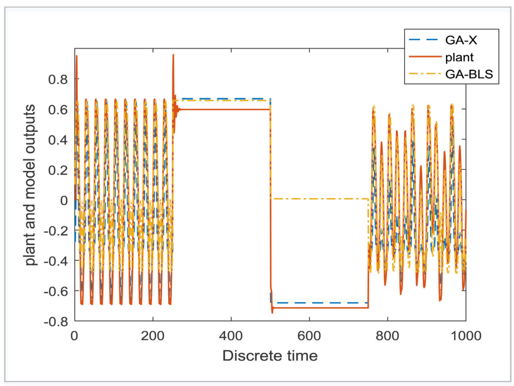

FIGURE 25. Modeling result of Example 4 based on GA-BLS and GA-X.

The BLS structure used in this article further strengthens the artificially expanded input information and directly maps to the output layer, avoiding the compression of information by the hidden layer, and uses a more effective expansion method for the input information, highlighting the advantages of BLS structures in identifying nonlinear systems. With the decrease in the signal-to-noise ratio, the convergence error of the proposed method is kept below 0.1, which leads to an advantage in anti-interference ability compared with the other three methods.

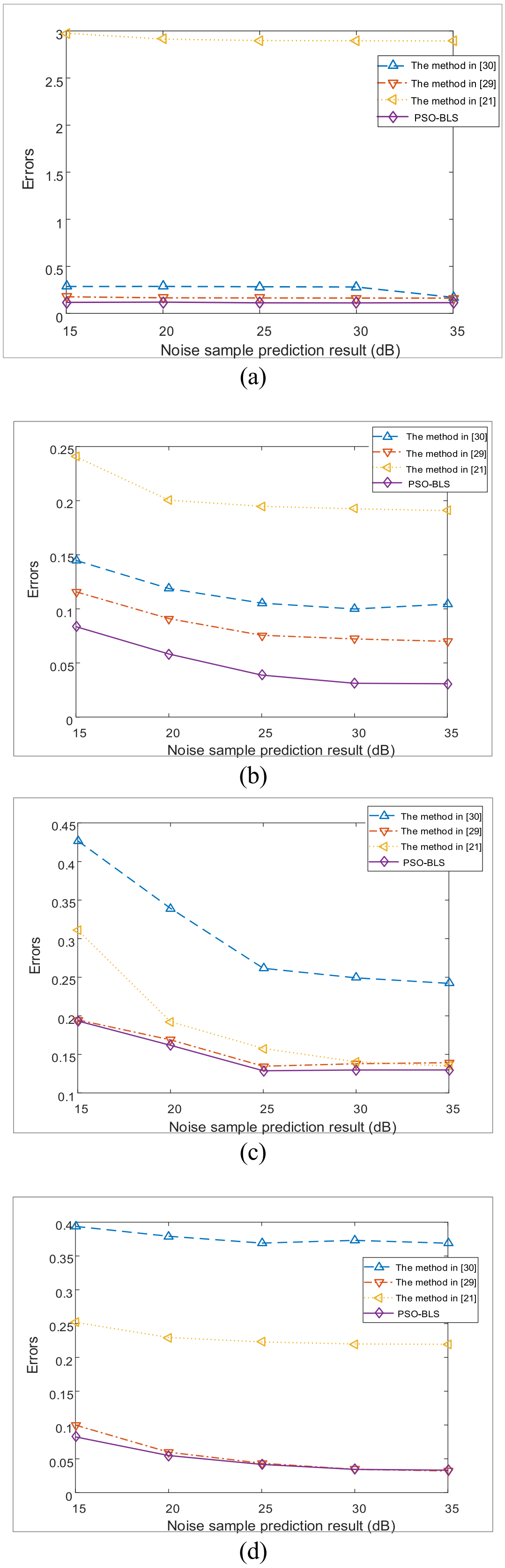

FIGURE 26. Comparison results obtained from the proposed method and the methods in [21], [29], [30]: (a) prediction results of example 1, (b) prediction results of example 2, (c) prediction results of example 3, (d) prediction results of example 4.

IV. CONCLUSION

In this paper, an algorithm combining a BLS and PSO is proposed to identify nonlinear dynamic systems. First, the output type of the system and the dimension of the input extension are determined and the model fitness function E is calculated by a BLS. Second, the relationship between the fitness function and the output weight is determined. Finally, PSO is used to optimize the output weight to find the global optimal solution. The simulation experiment shows that the proposed method has better prediction accuracy and anti-interference ability.

REFERENCES

[1] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[2] S.A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. West Sussex, U.K.: Wiley, 2013.

[3] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Berlin, Germany: Springer-Verlag, 2001.

[4] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Syst. Mag.*, vol. 39, no. 6, pp. 28–99, Dec. 2019.

[5] R. K. Pearson, B. A. Ogunnaike, and F. J. Doyle, "Identification of structurally constrained second-order volterra models," *IEEE Trans. Signal Process.*, vol. 44, no. 11, pp. 2837–2846, Nov. 1996.

[6] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[7] S. Formentin, M. Mazzoleni, M. Scandella, and F. Previdi, "Nonlinear system identification via data augmentation," *Syst. Control Lett.*, vol. 128, pp. 56–63, Jun. 2019.

[8] G. Pillonetto, A. Chiuso, and G. De Nicolao, "Prediction error identification of linear systems: A nonparametric Gaussian regression approach," *Automatica*, vol. 47, no. 2, pp. 291–305, Feb. 2011.

[9] H. Ohlsson, J. Roll, and L. Ljung, "Manifold-constrained regressors in system identification," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 1364–1369.

[10] B. O. Ayinde and J. M. Zurada, "Deep learning of constrained autoencoders for enhanced understanding of data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 3969–3979, Sep. 2018.

[11] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[12] D. Li, D. V. Vargas, and S. Kouichi, "Universal rules for fooling deep neural networks based text classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 2221–2228.

[13] K. Li, J. Kou, and W. Zhang, "Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple mach numbers," *Nonlinear Dyn.*, vol. 96, no. 3, pp. 2157–2177, May 2019.

[14] X. P. Zhu, J.-M. Dai, C.-J. Bian, Y. Chen, S. Chen, and C. Hu, "Galaxy morphology classification with deep convolutional neural networks," *Astrophys. Space Sci.*, vol. 364, no. 4, p. 55, Apr. 2019.

[15] N. Van Huynh, D. Thai Hoang, D. N. Nguyen, and E. Dutkiewicz, "Optimal and fast real-time resource slicing with deep dueling neural networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1455–1470, Jun. 2019.

[16] J. Wen, P.-F. Han, Z. Zhou, and X.-S. Wang, "Lake level dynamics exploration using deep learning, artificial neural network, and multiple linear regression techniques," *Environ. Earth Sci.*, vol. 78, no. 6, pp. 1–12, Mar. 2019.

[17] C. Chen and Z. Liu, "An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn.*, vol. 29, no. 1, pp. 1191–1204, Apr. 2018.

[18] L. Zhu, Z. Liu, C. L. P. Chen, and Y. Zhang, "A wavelet broad learning adaptive filter for forecasting and cancelling the physiological tremor in teleoperation," *Neurocomputing*, vol. 356, no. 3, pp. 170–183, Sep. 2019.

[19] M. J. Cui, M. Khodayar, C. Chen, X. Wang, Y. Zhang, and M. E. Khodayar, "Deep learning-based time-varying parameter identification for system-wide load modeling," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6102–6114, Nov. 2019.
[20] L. Murphy, C. Venkataraman, and A. Madzvamuse, "Parameter identification through mode isolation for reaction–diffusion systems on arbitrary geometries," *Int. J. Biomath.*, vol. 11, no. 4, Mar. 2018, Art. no. 1850053.
[21] M. Pelliciari, G. C. Marano, and T. Cuoghi, "Parameter identification of degrading and pinched hysteretic systems using a modified Bouc–Wen model," *Comput.-Aided Civil Inf.*, vol. 14, no. 12, pp. 1573–1585, May 2018.
[22] K. Yoon, D. Y. Kim, Y. C. Yoon, and M. Jeon, "Data association for multi-object tracking via deep neural networks," *Sensors*, vol. 19, no. 3, p. 559, Jan. 2019.
[23] J. Vörös, "Identification of nonlinear dynamic systems with input saturation and output backlash using three-block cascade models," *J. Franklin Inst.*, vol. 351, no. 12, pp. 5455–5466, Dec. 2014.
[24] U. Boz and M. Eriten, "Nonlinear system identification of soft materials based on Hilbert transform," *J. Sound Vib.*, vol. 447, pp. 205–220, May 2019.
[25] Y.-X. Zheng and Y. Liao, "Parameter identification of nonlinear dynamic systems using an improved particle swarm optimization," *Optik*, vol. 127, no. 19, pp. 7865–7874, Oct. 2016.
[26] H. Y. Khaw, F. C. Soon, J. H. Chuah, and C.-O. Chow, "High-density impulse noise detection and removal using deep convolutional neural network with particle swarm optimisation," *IET Image Process.*, vol. 13, no. 2, pp. 365–374, Feb. 2019.
[27] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1817–1824, Nov. 2008.
[28] H. Su, "Siting and sizing of distributed generators based on improved simulated annealing particle swarm optimization," *Environ. Sci. Pollut. Res.*, vol. 26, no. 18, pp. 17927–17938, Jun. 2019.
[29] N. S. Nguyen, V. Ho-Huu, and A. P. H. Ho, "A neural differential evolution identification approach to nonlinear systems and modelling of shape memory alloy actuator," *Asian J. Control*, vol. 20, no. 1, pp. 57–70, Jan. 2018.
[30] Y.-S. Yang, W.-D. Chang, and T.-L. Liao, "Volterra system-based neural network modeling by particle swarm optimization approach," *Neurocomputing*, vol. 82, pp. 179–185, Apr. 2012.
[31] J.-H. Seo, C.-H. Im, C.-G. Heo, J.-K. Kim, H.-K. Jung, and C.-G. Lee, "Multimodal function optimization based on particle swarm optimization," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1095–1098, Apr. 2006.
[32] G. Xu and G. Yu, "Reprint of: On convergence analysis of particle swarm optimization algorithm," *J. Comput. Appl. Math.*, vol. 340, pp. 709–717, Oct. 2018.
[33] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Inf. Process. Lett.*, vol. 85, no. 6, pp. 317–325, Mar. 2003.
[34] M. Jiang, Y. P. Luo, and S. Y. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Inf. Process. Lett.*, vol. 102, no. 1, pp. 8–16, Apr. 2007.
[35] H. Zhao and J. Zhang, "Nonlinear dynamic system identification using pipelined functional link artificial recurrent neural network," *Neurocomputing*, vol. 72, nos. 13–15, pp. 3046–3054, Aug. 2009.

RAN HAN received the B.S. degree in electrical engineering and automation from Shanghai Ocean University, China, in 2018. He is currently pursuing the M.S. degree in naval and ocean engineering with Jimei University. His research interests include fault diagnosis of power electronics circuits and intelligent information processing.

RONGJIE WANG received the Ph.D. degree in electrical and electronic engineering from Sun Yat-sen University, China, in 2012. He is currently a Professor and a vice-supervisor of Ph.D. students with the Marine Engineering Institute, Jimei University, China. His research interests include system identification, fault diagnosis of power electronics circuits, and blind source separation.

GUANGMIAO ZENG received the B.S. degree in ship electrical and electronic engineering from Jimei University, China, in 2018, where he is currently pursuing the M.S. degree in naval and ocean engineering. His research interests include fault diagnosis of power electronics circuits and intelligent information processing.