Hindawi
Complexity
Volume 2021, Article ID 8086088, 12 pages
https://doi.org/10.1155/2021/8086088

Research Article

Fuzzy Wavelet Neural Network with the Improved Levenberg–Marquardt Algorithm for the AC Servo System

Run-Min Hou, Di-Fen Shi, Qiang Gao, and Yuan-Long Hou

School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Correspondence should be addressed to Run-Min Hou; 187189579@qq.com

Received 13 August 2021; Revised 13 September 2021; Accepted 11 October 2021; Published 23 October 2021

Academic Editor: Long Wang

In this study, a fuzzy wavelet neural network with the improved Levenberg–Marquardt algorithm (FWNN-LM) is proposed to conquer nonlinearity and uncertain disturbance problems in the AC servo system. First of all, use the particle swarm optimization algorithm based on Levenberg–Marquardt (LM) to optimize parameters in the FWNN controller. Second, the potentiality of fuzzy rules (PFR) method is developed to optimize the structure of the FWNN by error reduction ratio (ERR). Furthermore, stability of FWNN-LM is proved by the Lyapunov method. Finally, simulation and prototype test results show that this method can improve the accuracy and robustness of the system in presence of load disturbances and parameter perturbations.

1. Introduction

In recent years, varies studies show that the AC servo system exhibits good dynamical property [1, 2], but the stability still needs to improve. For the AC servo system, the dynamic mathematical model is a complex system with characteristics of large load, which can lead to nonlinearity and uncertain disturbance. In practical applications, an AC servo system performance may be affected due to unmodeled dynamics changed [3, 4].

After referring to many references, a lot of studies have shown the neural network is an important component of a complicated nonlinear system control policy under the circumstance of the lack of full model details [5–8]. The most prominent advantage of the neural network is approximate capability, and it can approximate function with any precision. However, it is hard to avoid local minimization for the BP neural network. If we use a sigmoid function as the stimulation function, it also causes slow convergence speed. Moreover, it cannot realize the mapping rules in time [9]. Fuzzy logic has become a hot topic research of neural networks in many studies. Dong et al. [10, 11] provide theoretical basis to modelling and controlling the nonlinear system. Consider there are many uncertainties existing in the fuzzy control process. Wang [12] provides a fuzzy neural network along with utilization to improve system robustness without accurate control; however, the parametric learning algorithm is presupposed for the topology of fuzzy systems.

As an alternative, multiple research studies concentrate on the use of the wavelet neural network (WNN) [13–16]. Compared with the usual sigmoid function neural network, the wavelet function possesses a better learning capacity in aspects of system identification. In recent years, Zekri et al. [17–19] studied the combination of wavelet theory and the fuzzy neural network (FNN). In the FWNN, fuzzy rules are corresponding to the sub-WNN, respectively, and the wavelet and fuzzy sets parameters learning can improve the FWNN approximation accuracy [20–23]. However, the main drawback of the WNN is that due to its feed-forward network structure, its application area is limited to static issues. The Levenberg–Marquardt (LM) method has a remarkable characteristic of local learning and a fast convergence performance at the same time [24]. However, the LM algorithm increases memory demands with the method of calculating some problems that come from the error function with the Jacobian matrix [25]. Moreover, another disadvantage is that the LM algorithm is still a local optimization method.

The particle swarm algorithm (PSO) is a global optimization algorithm, through collaboration and competition between individuals to find the optimal solution, and the particle swarm optimization search process is started from the entire group, with the implicit parallel search features to improve the performance of the algorithm [22]. However, the PSO algorithm has some disadvantages such as slow convergence speed.

Based on the above analysis, in this study, an adaptive fuzzy wavelet neural network controller with LM is proposed to control the rotor position of the AC servo system for tracing reference trajectory with robustness. In the proposed control structure, the FWNN is a controller, and the LMPSO algorithm is employed for the online training of all weights of the FWNN. Moreover, potentiality of fuzzy rules (PFR) with using error reduction ratio (ERR) is developed to adjust the parameters and organize the structure of the FWNN. The stability of the system can be proved by using Lyapunov theory [26]. Finally, studies demonstrate promising results of a prototype AC servo system that can verify the feasibility and effectiveness by using the proposed algorithm.

The contents of this study can be listed as follows: the second section analyzes the servo system. After briefly introducing the FWNN in the third section, the following section four develops the FWNN-LM which has been proposed at great length. Afterwards, the convergence of the algorithm is analyzed in section five. And then, the simulation outcomes are discussed in section six. Last, a conclusion has been mentioned in the last section.

2. AC Servo System Analysis

The AC servo system control structure is shown in Figure 1.

In the stationary (d-q) frame of reference, the mathematical models of the permanent magnet synchronous motor can be expressed as follows:

$$\begin{cases} \dot{i}_d = -\dfrac{R}{L_d} i_d + \dfrac{L_q}{L_d} p i_q \omega_r + \dfrac{u_d}{L_d}, \\ \dot{i}_q = -\dfrac{R}{L_q} i_q + \dfrac{L_d}{L_q} p i_d \omega_r - \dfrac{\psi_f}{L_q} p \omega_r + \dfrac{u_q}{L_q}, \\ \dot{\omega}_r = \dfrac{1}{J} \left(T_e - T_L - B\omega_r \right), \end{cases} \quad (1)$$

where i_d, i_q, u_d, u_q, and L_d, L_q represent the electric currents, voltages, and inductance coefficient of the motor d and q axes, respectively; R is the motor stator resistor (Ohm), ψ_f represents the motor permanent magnet flux, p represents the motor pair of poles, J represents the motor inertia constant, B represents the viscous friction coefficient, ω_r stands for the motor angular velocity, T_e stands for motor electromagnetic torque, and T_L stands for the load torque.

The system is applied to a three-closed-loop control system. It uses the magnetic field-oriented control technology to complete the motor position and achieve high performance. Additionally, the simplification of the motor control system uses the $i_d = 0$ vector control approach. When $i_d = 0$, the motor mechanical equation can be expressed as

$$J\dot{\omega}_b + B\omega_b + T_L = T_e, \quad (2)$$

where ω_b is the mechanical angular velocity, and T_e can be written as

$$T_e = \frac{3}{2} p \psi_f i_q = K_t i_q, \quad (3)$$

where K_t is a moment constant that needs to be adjusted.

Generally, compared with the mechanical time constant, the motor current time constant has a much smaller numerical value; thus, the delay time of the current responding can be neglected. The state variables can be set as $x_1 = \theta$ and $x_2 = \dot{\theta}$; substitute equation (2) into equation (3), and the AC servo system can be rewritten as

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -\dfrac{B}{J} x_2(t) + \dfrac{K_t}{J} i_q(t) + \left(-\dfrac{1}{J} T_L \right), \end{cases} \quad (4)$$

where $-(B/J)$, K_t/J, and $-(1/J)T_L$ represent the nonlinear dynamic equations and the external disturbance, respectively.

3. Fuzzy Wavelet Neural Network

3.1. Wavelet Neural Network Structure. The structure of wavelet neural network is shown in Figure 2. As Figure 2 illustrates, K is the master nodes of the input layer, the hidden layer number is M, ω_{km} is the connection weighing between node k of the input layer and node m of the hidden layer, ω_m is the connection weighing between node m and the output layer, b_m is the translation parameter of wavelet function, and a_m is the scale variable of the wavelet function. The output can be written as [21]

$$\mathbf{L}(t) = \sum_{m=1}^{M} \boldsymbol{\omega}_m \boldsymbol{\psi}_m \left(\text{net}_m \right), \quad (5)$$

where $\text{net}_m = \sum_{k=1}^{K} \omega_{km} x_k - b_m / a_m$, $(m = 1, 2, \ldots, M)$. Choose the Morlet wavelet function as the generating function $\psi(x) = \cos(1.75x) \exp(-(1/2)x^2)$.

3.2. Fuzzy Wavelet Neural Network Structure. In the fuzzy wavelet network, each fuzzy rule corresponds to a given wavelet scale values of the wavelet neural network [27]. In order to describe FWNN-LMPSO clearly, a simple structure of the FWNN is shown in Figure 3.

The N_F fuzzy IF-THEN rules can be expressed as follows:

R_n: if x_1 is A_{1n},x_2 is A_{2n}, ..., and x_m is A_{mn}, then

$$\mathbf{y}_n = \mathbf{L}_n = \sum_{m=1}^{N_{w,n}} \boldsymbol{\omega}_{m,n} \boldsymbol{\psi}_{m,n}(x), \quad (6)$$

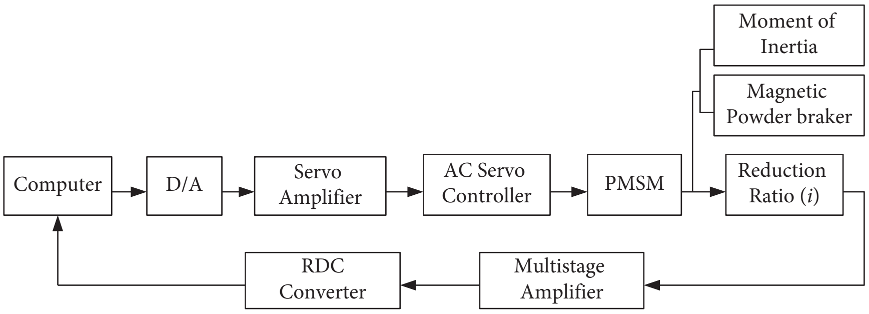

Figure 1: The structure of the AC servo system.

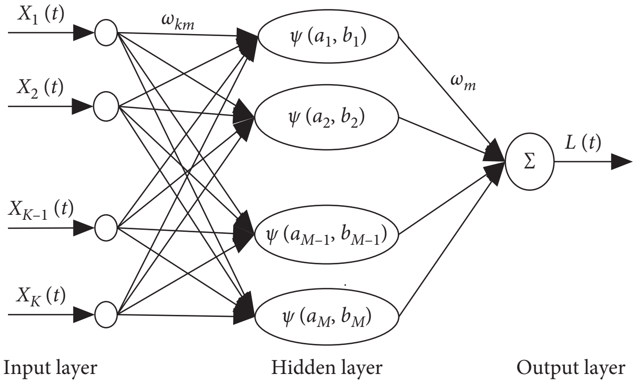

Figure 2: The structure of the wavelet neural network.

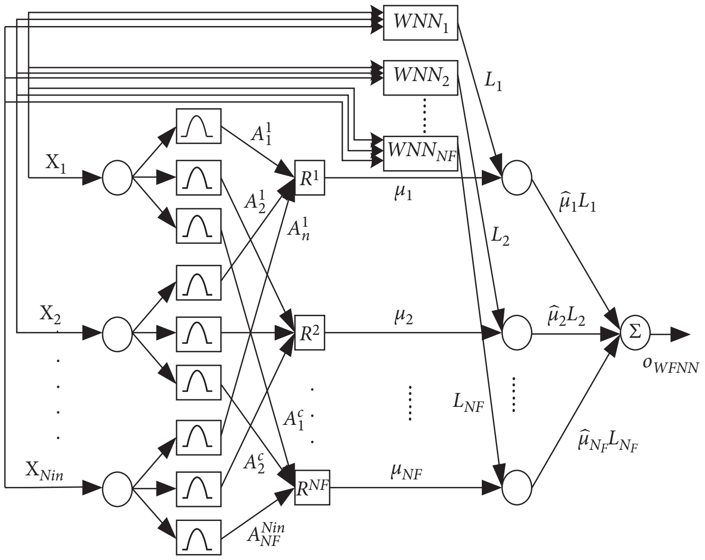

Figure 3: The structure of the FWNN.

where R_n is the fuzzy rule $(1 \leq n \leq N_F)$; A_{mn} is the membership function for the fuzzy set of Gaussian function, which can be expressed as

$$\mu_{A_{mn}}(x_m) = \exp^{\left(-(x_m - c_{mn})^2/\sigma_{mn}^2\right)}, \tag{7}$$

where x_m is the input of $m = 1{:}\ N_{in}$, N_{in} represents the number of input neurons; $n = 1{:}\ N_F$. The canter c_{mn} and width σ_{mn} can be used to define as a subordinate function.

The output of the entire FWNN structure by using product rules and defuzzification is shown as

$$\mathbf{O}_{\mathrm{FWNN}}(k) = \sum_{n=1}^{N_F} \widehat{\mu}_n(x)\mathbf{L}_n, \tag{8}$$

where $\widehat{\mu}_n(x) = (\mu_n(x)/\sum_{n=1}^{N_F} \mu_n(x))$, $\mathbf{L}_n = \sum_{j=1}^{N_{W(n)}} \omega_j \psi_j$, and $\mu_n(x) = \prod_m \mu_{A_{mn}}(x_m)$.

The output mean square error of the online learning is

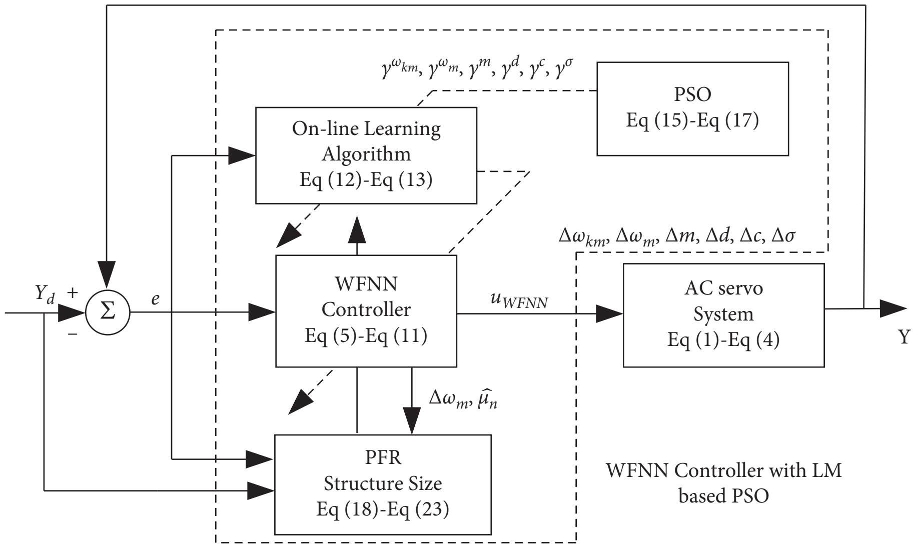
$\gamma^{\omega_{km}}, \gamma^{\omega_m}, \gamma^m, \gamma^d, \gamma^c, \gamma^\sigma$
PSO
Eq (15)-Eq (17)
On-line Learning
Algorithm
Eq (12)-Eq (13)
$\Delta\omega_{km}, \Delta\omega_m, \Delta m, \Delta d, \Delta c, \Delta\sigma$
Y_d
+
−
Σ
e
WFNN
Controller
Eq (5)-Eq (11)
u_{WFNN}
AC servo
System
Eq (1)-Eq (4)
Y
$\Delta\omega_m, \widehat{\mu}_n$
PFR
Structure Size
Eq (18)-Eq (23)
WFNN Controller with LM
based PSO

Figure 4: FWNN with LM-based PSO.

$$E = \frac{1}{2}\left[\left(O_d(k) - O_{\mathrm{FWNN}}(k)\right)^2\right], \tag{9}$$

where O_d is the expected output of the training data. According to the descent algorithm, the FWNN parameters adjustment formulas are shown as

$$\begin{gathered} \omega_{km}^* = \omega_{km} - \gamma^1\frac{\partial E}{\partial\omega_{km}};\ \omega_m^* = \omega_m - \gamma^2\frac{\partial E}{\partial\omega_m};\ a_m^* = a_m - \gamma^3\frac{\partial E}{\partial a_m}, \\ b_m^* = b_m - \gamma^4\frac{\partial E}{\partial b_m};\ c_{mn}^* = c_{mn} - \gamma^5\frac{\partial E}{\partial c_{mn}};\ \sigma_{mn}^* = \sigma_{mn} - \gamma^6\frac{\partial E}{\partial\sigma_{mn}}, \end{gathered} \tag{10}$$

where $\gamma^l\,(l = 1{:}\ 6)$ is the learning rate, and the arguments of the FWNN controller can be expressed as

$$\boldsymbol{\gamma}^l = \left[\gamma^1, \ldots, \gamma^2, \ldots, \gamma^l\right] = \left[\gamma^{\omega_{km}}, \gamma^{\omega_m}, \gamma^m, \gamma^d, \gamma^c, \gamma^\sigma\right]. \tag{11}$$

4. FWNN with LM Algorithm-Based PSO

In the neural network, a sigmoid function is used as the activation function of the BP neural network, which leads to the result that the BP neural network is easy to get into a local minimum, slow convergence speed. Thus, to improve the performance of FWNN, the training process is required to adjust both the structure size and the parameters. An LMPSO method is used for adjusting the parameters; and a PFR is developed to design the structure of FWNNs. In the following, the LMPSO method and PFR method are described in detail.

4.1. LM Algorithm-Modified FWNN. The LM algorithm is an approximate Newton algorithm, which proves that the LM-based BPNN algorithm converges quick and accurate performance [28]. In this study, the total mean square error of **P** is given as

$$\mathbf{P} = \frac{1}{2}\sum_{p=1}^{P} e_p^2. \tag{12}$$

The LM algorithm is written as

$$\Delta\,\mathbf{h} = \left[\mathbf{J}^T(h)J(h) + \lambda I\right]^{-1}\mathbf{J}^T(h)\bar{\mathbf{e}}(h), \tag{13}$$

where the Jacobian matrix $\mathbf{J}(h)$is

$$\mathbf{J}(h) = \begin{bmatrix} \dfrac{\partial e_1(h)}{\partial h_1} & \dfrac{\partial e_2(h)}{\partial h_2} & \cdots & \dfrac{\partial e_1(h)}{\partial h_I} \\ \dfrac{\partial e_1(h)}{\partial h_1} & \dfrac{\partial e_2(h)}{\partial h_2} & \cdots & \dfrac{\partial e_2(h)}{\partial h_I} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \dfrac{\partial e_P(h)}{\partial h_1} & \dfrac{\partial e_P(h)}{\partial h_2} & \cdots & \dfrac{\partial e_P(h)}{\partial h_I} \end{bmatrix}, \tag{14}$$

$$\bar{\mathbf{e}}(h) = \left[e_1, e_2, \ldots, e_P\right]^T.$$

The structure of the fuzzy wavelet neural network (FWNN) based on the LMPSO controller is shown in Figure 4.

4.2. LM Algorithm-Based PSO. PSO is a population-based heuristic global optimization technique. In this algorithm, the population is called a swarm, and the trajectory of each particle in the search space is adjusted by dynamically altering its velocity, according to its own flying experience and swarm experience in the search space. In the PSO algorithm, a group of particles represent a candidate solution. The velocity and position updating formulas of the PSO are illustrated as

$$\begin{cases} v_i(k+1) = w(k)v_i(k) + c_1r_1\left(P_i(k) - x_i(k)\right) + c_2r_2\left(P_g(k) - x_i(k)\right), \\ x_i(k+1) = x_i(k) + v_i(k+1), \end{cases} \tag{15}$$

where $v_i(k)$ represents the current rate of particle i^{th} during the iteration k; $x_i(k)$ represents the current position of the i^{th} particle; $P_i(k)$ is on behalf of the optimum position of the i^{th} particle previously appeared; $P_g(k)$ denotes the best previous position among all the particles; c_1 and c_2 are on behalf of the acceleration factors; r_1 and r_2 uniformed random number in the interval $[0,1]$; $w$stands for the inertia weight in the interval $[0.4, 0.9]$. An appropriate fitness function to calculate the appropriate value is

$$F(t) = 0.4e_{\mathrm{MAE}} + 0.6e_{\max}, \tag{16}$$

where $F(t)$ is the fitness value; e_{MAE} represents the mean absolute error, $e_{\max}$ is the maximum absolute error, and connected with LM-based FWNN, there is

$$e_{\mathrm{MAE}} = \frac{\sum_{x=1}^{X}\left|e_x\right|}{X};\ e_{\max} = \max\left(\left|\bar{e}(h)\right|\right);\ x_i = h. \tag{17}$$

In this study, the PSO-LM algorithm is used to make adjustments of FWNN which can make e_{MAE} and $e_{\max}$ more appropriate in actual conditions. A few particulars about the PSO procedure are shown as follows:

(1) Initialize the PSO parameters

(2) Determine $P_g(k)$ and $P_i(k)$

(3) Refresh the particle speed as well as the position by taking advantage of equation (29)

(4) Get the current fitness value FT and update$P_i(k)$, $P_g(k)$; if $\mathrm{FT}_i < P_i(k)$, $\mathrm{FT}_i < P_g(k)$.

(5) If$i < N$, set $i = i + 1$ and then go to step (3); otherwise, proceed to the next step.

(6) If $k <$ maxgen, set $k = k + 1$ and then go to step (2) or output $P_g(k)$ to LM-based FWNN.

4.3. Potentiality of Fuzzy Rules (PFR). The PFR values can be used to calculate the potentiality of fuzzy rules and extract the contributions of the normalized neuron. The FWNN model is expressed as follows:

$$\mathbf{y}_d(t) = \mathbf{W}(t)\boldsymbol{\Phi}(t) + \mathbf{e}(t), \tag{18}$$

where $\mathbf{W}(t) = [\mathbf{w}(t-k-1), \mathbf{w}(t-k+2), \ldots, \mathbf{w}(t)]^{\mathrm{T}}$ is the weight between the output layer and normalized layer, and $\Phi(t)$ is given by

$$\boldsymbol{\Phi}(t) = [\widehat{\mu}(t-k+1), \widehat{\mu}(t-k+2), \ldots, \widehat{\mu}(t)]. \tag{19}$$

The matrix $\Phi(t)$ can be transformed into a set of orthogonal basis vectors by QR decomposition as

$$\boldsymbol{\Phi}^{\mathrm{T}} = \mathbf{Q}(t)\mathbf{R}(t), \tag{20}$$

where $R(t)$ is an upper triangular matrix, and $\mathbf{Q}(t) = [q_1(t), q_1(t), \ldots, q_n(t)]$ have the same dimension as $\Phi(t)$. Then, the ERR is given by [29]

$$\operatorname{error}(t) = \frac{\left(\mathbf{y}_d(t)\mathbf{q}_l^{\mathrm{T}}(t)\right)^2}{\mathbf{q}_l^{\mathrm{T}}(t)\mathbf{q}_l(t)\mathbf{y}_d(t)\mathbf{y}_d^{\mathrm{T}}(t)}, \quad l = 1, 2, \ldots, N_F. \tag{21}$$

The PFR value of the l_{th} normalized neuron can be expressed as follows:

$$\mathrm{PFR}_l(t) = \frac{R_l(t)}{\sum_{l=1}^{N_F} R_l(t)}, \quad l = 1, 2, \ldots, N_F, \tag{22}$$

where $\mathrm{PFR}_l(t) \in (0,1)$ is the potentiality of fuzzy rule in the l^{th} normalized neuron, and

$$R_l(t+1) = \eta R_l(t) + \widehat{\mu}(t-\rho+1)\operatorname{error}(t), \quad \rho = k, k-1, \ldots, 1, \tag{23}$$

where $0 < \eta < 1$ is a constant.

5. Stability Analysis

Lyapunov function is used to assess the system stability, and it can be defined as

$$V(k) = \frac{1}{2}e^2(k), \tag{24}$$

where $e(k) = (y_d(k) - y(k)).y_d$ is the desired output, and $y(k)$ is the actual output.

$$\Delta V(k) = \frac{1}{2}\left(e^2(k+1) - e^2(k)\right), \tag{25}$$

where $e(k+1) = e(k) + \Delta e(k)$. Using Taylor's formula, $\Delta e(k)$ can be given as

$$\Delta e(k) = \sum_{n=1}^{N_F}\left\{\left[\frac{\partial e(k)}{\partial \mathbf{P}_n^l(k)}\right]^T \Delta \mathbf{P}_n^l(k)\right\}, \tag{26}$$

where

$$\begin{cases} \dfrac{\partial e(k)}{\partial P_n^l(k)} = -A \dfrac{\mu_n(x)}{\sum_{h=1}^{N_F} \mu_h(x)} \dfrac{\partial L_n}{\partial P_n^l(k)}, \\ \Delta P_n^l(k) = \gamma^l A e(k) \times \left(\dfrac{\mu_n(x)}{\sum_{h=1}^{N_F} \mu_h(x)} \right) \dfrac{\partial L_n}{\partial P_n^l(k)}, \\ \mathbf{P}_n = \left[P_n^1, \ldots, P_n^2, \ldots, P_n^l \right] = \left[\omega_{Km,(n)}, \omega_{m,(n)} m_{m,(n)}, d_{m,(n)}, c_{m,(n)}, \sigma_{m,(n)} \right], \\ n = 1\colon N_F, l = 1\colon 6, j = 1\colon m, k = 1\colon N_{\text{in}}, \end{cases} \tag{27}$$

where $A = (\partial y(k)/\partial u(k))$. Then, $\Delta e(k)$ can be written as

$$\Delta e(k) = \gamma^l A^2 e_1(k) \frac{1}{\left(\sum_{h=1}^{N_F} \boldsymbol{\mu}_h(x)\right)^2} \sum_{n=1}^{N_F} \boldsymbol{\mu}_n^2(x) \left\| \frac{\partial \mathbf{L}_n}{\partial \mathbf{P}_n^l(k)} \right\|^2. \tag{28}$$

Substituting equation (28) into equation (25), $\Delta V(k)$ can be rewritten as

$$\Delta V(k) = -\frac{1}{\left(\sum_{h=1}^{N_F} \boldsymbol{\mu}_h(x)\right)^2} \times \sum_{n=1}^{N_F} \left(\boldsymbol{\mu}_n^2(x) \left\| \frac{\partial \mathbf{L}_n}{\partial \mathbf{P}_n^l(k)} \right\| \right)^2 A e_1(k) \lambda, \tag{29}$$

where

$$\lambda = \gamma^l \left[1 - \frac{1}{2} \frac{\gamma^l}{\left(\sum_{h=1}^{N_F} \boldsymbol{\mu}_h(x)\right)^2} \times \sum_{n=1}^{N_F} \left(\boldsymbol{\mu}_n^2(x) \left\| \frac{\partial \mathbf{L}_n}{\partial \mathbf{P}_n^l(k)} \right\|^2 \right) \right]. \tag{30}$$

If $\mu_n(x) \leq 1$, that is,

$$\lambda \geq \gamma^l \cdot \left[1 - \frac{1}{2} \frac{\gamma^l}{\left(\sum_{h=1}^{N_F} \boldsymbol{\mu}_h(x)\right)^2} \times \sum_{n=1}^{N_F} \left(\left\{ \max\left(\boldsymbol{\mu}_n^2(x) \right) \right\}^2 \left\{ \max \left\| \frac{\partial \mathbf{L}_n}{\partial \mathbf{P}_n^l(k)} \right\| \right\}^2 \right) \right]. \tag{31}$$

Thus, according to Lyapunov stability theory, when $\lambda > 0$, $\Delta V(k) < 0$, and the stability of the system will be guaranteed.

6. Simulation Test and Discussion

In order to test and verify the effectiveness of the FWNN-LMPSO control, it will be compared with FWNN-LM.

6.1. Simulation Experiment. In addition, all simulation programs are conducted in Matlab/Simulink, and a clock rate of 2.6 GHz and 4 GB of RAM on a PC running in a Microsoft 7.0 environment are selected. The main parameters of the AC servo system are given in Table 1. Figures 5–9 show the simulation results.

As shown in Figure 5, the moment of inertia changes from the initial value to 1.5 times. FWNN-LM generates an overshoot; it takes 4.15 s to reach the stable condition. Using FWNN-LMPSO control, the system responds quickly, and only needs 1.6 s to reach the steady state without overshoot.

Figure 6 shows step response when a 360 nm disturbance added at 3 s.

As Figure 6 shows, when the load added, it gets more deviation results on account of the response of the algorithm of FWNN-LM control. It also has a 5.15° delay in tracking the reference position. However, when using the FWNN-LMIPSO control algorithm, the offset can decrease to 1.25°. It costs 0.35 s to reach the target position. Above all, the system can perform better in the aspect of load disturbance suppressing.

Experimental result of tracking step signal with random disturbances is shown in Figure 7. It can be seen from Figure 7 that when adding random disturbance to the response signal, there is no offset occurring by using FWNN-LMPSO control. Moreover, random disturbance is also added in sinusoidal tracking experiment. The maximum error of FWNN-LM and FWNN-LMPSO is 0.089° and 0.057°, respectively.

The sinusoidal tracking error curves with a frequency of 1.67 rad/s and amplitude of 30 degree is shown in Figure 8.

In Figure 9, the number of FWNN-LM iterations is about 220 steps, the training error is 0.128, and the training error of FWNN-LMPSO is 0.035 when the number of iterations is about 95 steps. Therefore, the convergence rate of FWNN-LMPSO is better than the FWNN-LM method.

6.2. Semiphysical Experiment. The semiphysical experiment platform structure is shown in Figure 10. A step response with FWNN-LM control and FWNN-LMPSO control are

Table 1: The main parameters of the AC servo system.

System parameters	S	Value	Unit
Converted to the motor output shaft moment of inertia	J	5239	$\mathrm{Kg.m}^2$
Converted to unbalanced torque and friction torque of the motor output shaft	T_L	9.32×10^3	$\mathrm{N \cdot m}$
Electromagnetic torque coefficient	K_t	0.195	$\mathrm{N \cdot m/A}$
Viscous friction coefficient	B	1.43×10^{-4}	$\mathrm{N \cdot m/(rad.s^{-1})}$
Reduction ratio	i	1039	

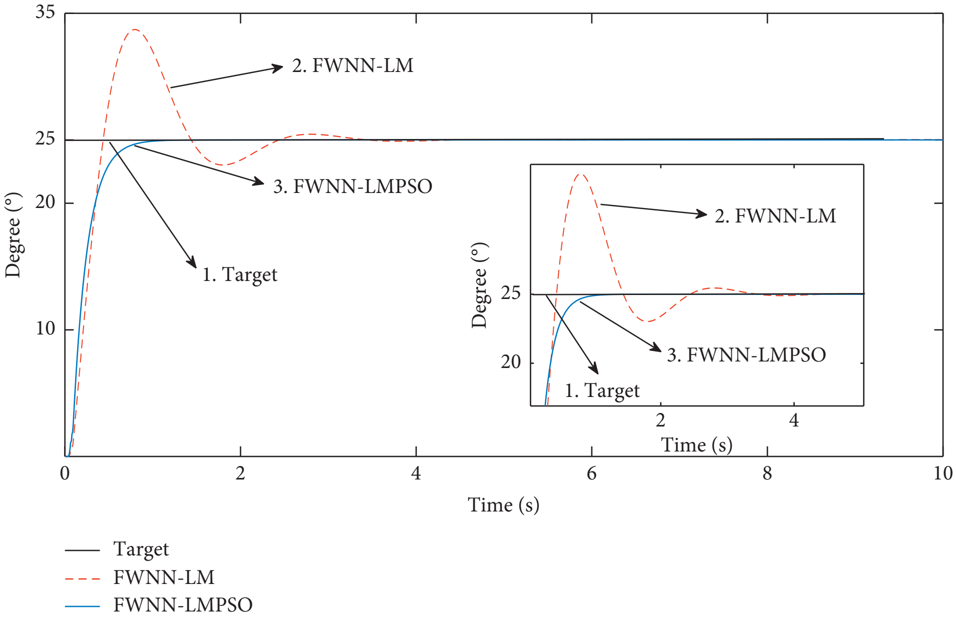

Figure 5: The moment of inertia changes the step response curve.

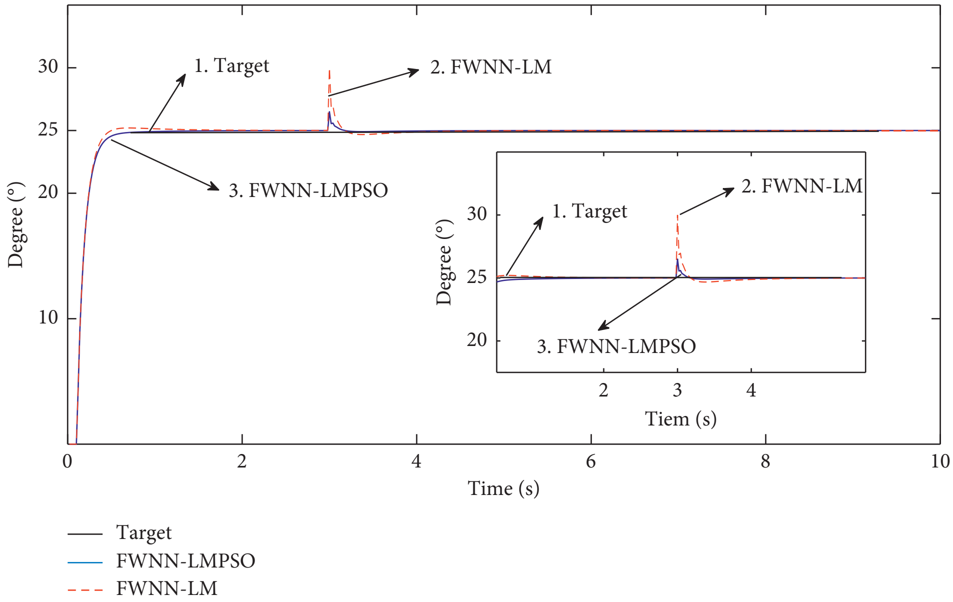

Figure 6: Step response curve of load disturbance.

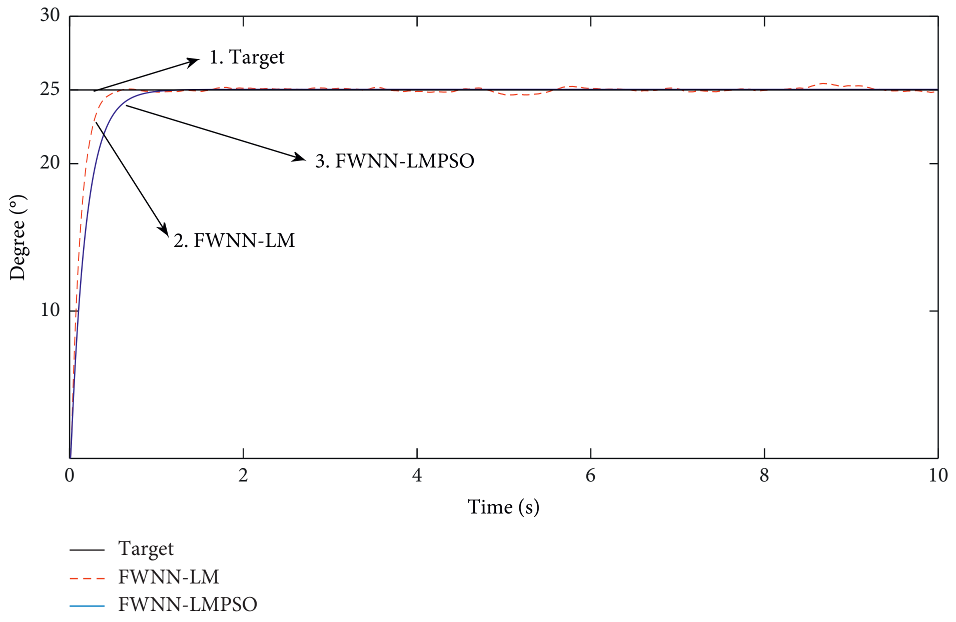

Figure 7: System dynamic response curves.

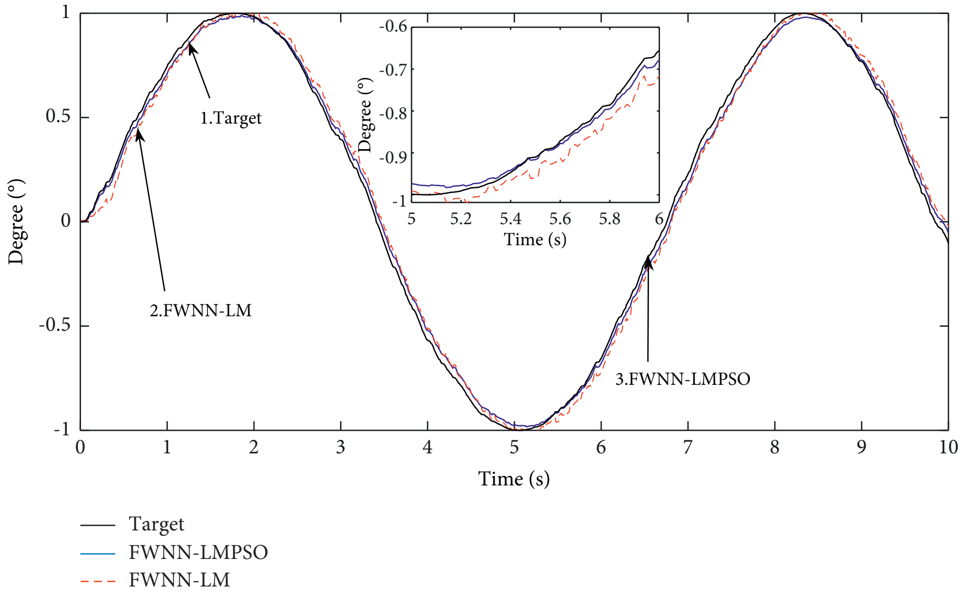

Figure 8: Sinusoidal tracking error curves with a neural network controller.

conducted on this semiphysical experiment platform to test the system performance.

According to Figures 11–14, when the system is under maximum load, its steady state time takes 1.41 s for the FWNN-LM control and the maximum steady state error is 2.63°; however, for FWNN-LMPSO control, the system required steady state time is 1.35 s, and maximum steady state error is 0.749°.

Compared with FWNN-LM control, the FWNN-LMSPSO control has better dynamic and steady state performance. In addition, it performs well in improving system antidisturbance performance.

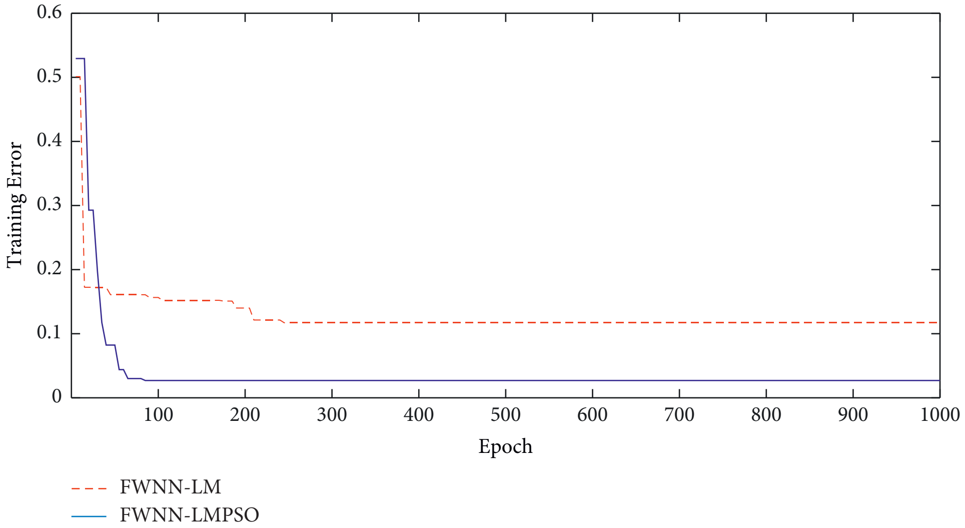

Figure 9: RMSE values in the training process.

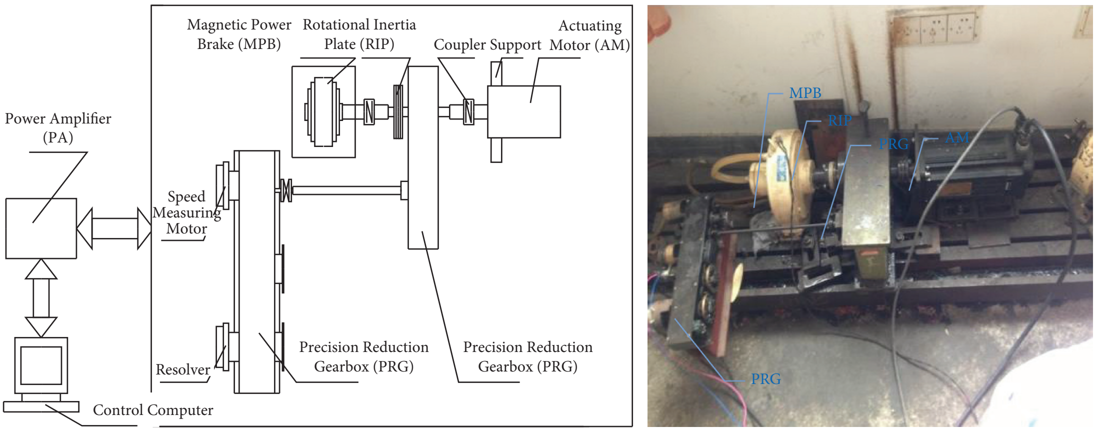
Magnetic Power Brake (MPB)
Rotational Inertia Plate (RIP)
Coupler Support
Actuating Motor (AM)
Power Amplifier (PA)
Speed Measuring Motor
Resolver
Precision Reduction Gearbox (PRG)
Precision Reduction Gearbox (PRG)
Control Computer
MPB
RIP
PRG
AM
PRG

Figure 10: Semiphysical platform structure.

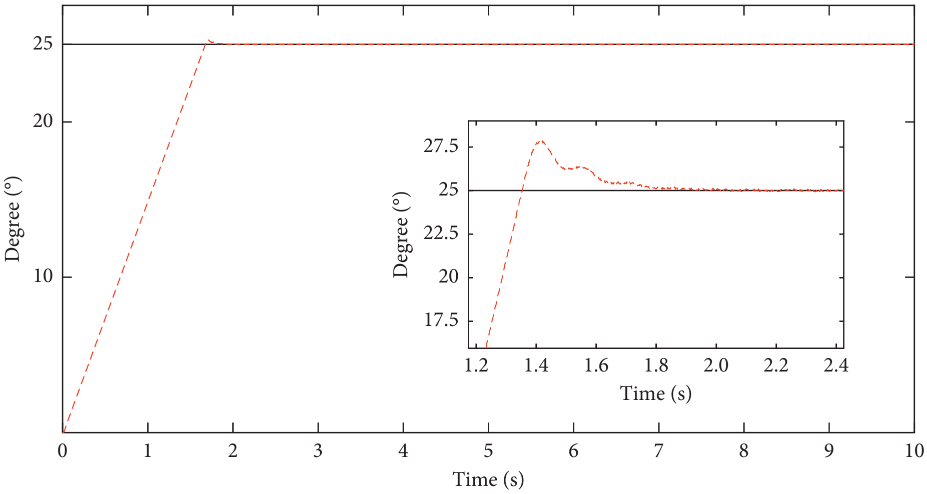

Figure 11: FWNN-LM step response curves with maximum load.

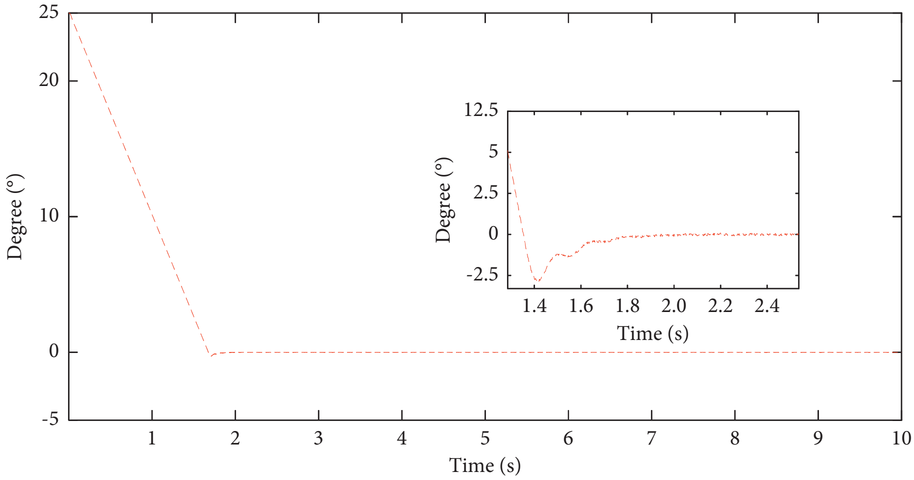

Figure 12: FWNN-LM error curves with maximum load.

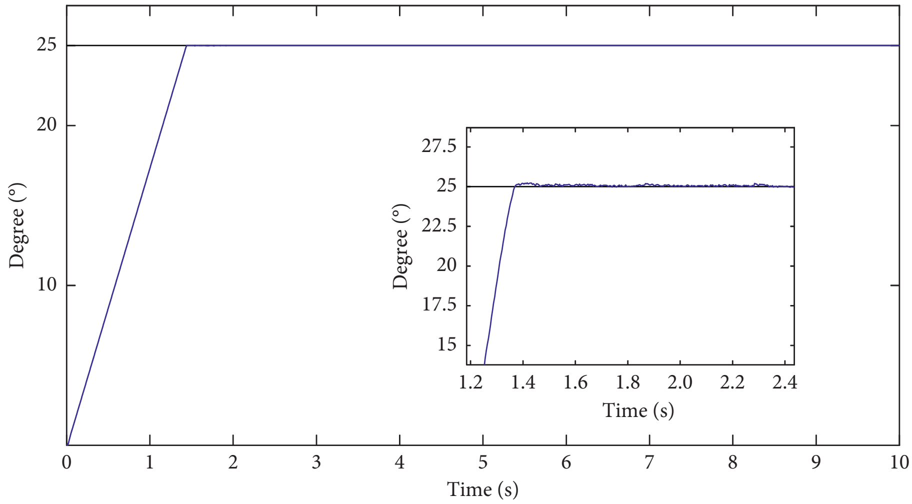

Figure 13: FWNN-LMPSO step response curves with maximum load.

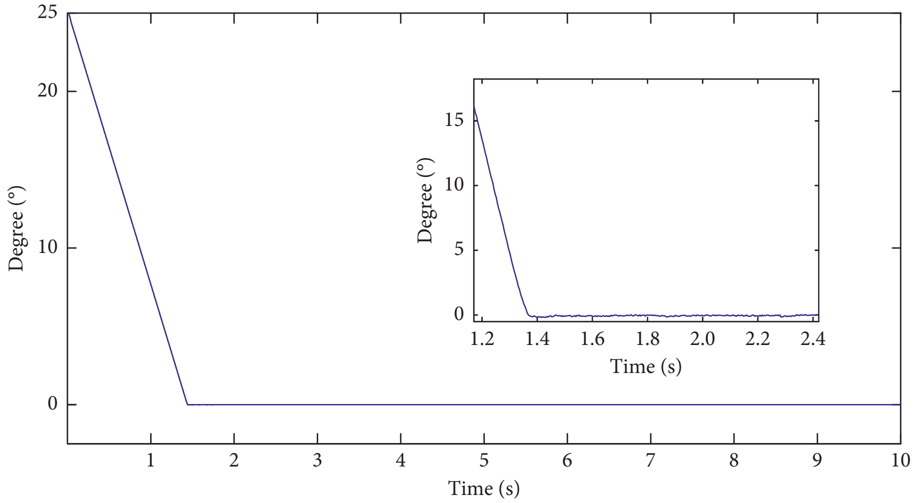

Figure 14: FWNN-LMPSO error curves with maximum load.

7. Conclusions

This study offers a new fuzzy wavelet neural network method in the AC servo system. Compared with the FWNN-LM controller, the proposed FWNN-LMPSO controller can be designed more accurately, more meaningful, and simpler. The main advantages of the existing method based on FWNN-LMPSO are as follows: first, in the FWNN-LMPSO based on the PFR method, fuzzy rules can be added to and removed from the structure learning method with the method of using the ERR value. Second, the LM algorithm improves the control accuracy through the adjustment of parameters, and the PSO learning algorithm is used to improve the learning speed. Lyapunov theory is also introduced to analysis system stability. Last, experimental results show the method has strong robustness and better dynamic performance.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (51805264).

References

[1] S.-M. Yang and K.-W. Lin, "Automatic control loop tuning for permanent-magnet AC servo motor drives," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1499–1506, 2016.

[2] L. Liu, T. Gao, Y.-J. Liu, S. Tong, C. L. P. Chen, and L. Ma, "Time-varying IBLFs-based adaptive control of uncertain nonlinear systems with full state constraints," *Automatica*, vol. 129, no. 2021, Article ID 109595, 2021.

[3] S. Zhigang Liu and Z. Liu, "Adaptive speed control for permanent-magnet synchronous motor system with variations of load inertia," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 8, pp. 3050–3059, 2009.

[4] K. Jezernik and M. Rodic, "High precision motion control of servo drives," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3810–3816, 2009.

[5] M. Sarma and K. K. Sarma, "An ANN based approach to recognize initial phonemes of spoken words of assamese language," *Applied Soft Computing*, vol. 13, no. 5, pp. 2281–2291, 2013.

[6] L. Liu, X. Li, Y.-J. Liu, and S. Tong, "Neural network based adaptive event trigger control for a class of electromagnetic suspension systems," *Control Engineering Practice*, vol. 106, no. 2021, Article ID 104675, 2021.

[7] O. Khayat, J. Razjouyan, F. N. Rahatabad, and F. Nowshiravan Rahatabad, "A fast learnt fuzzy neural network for huge scale discreted at a function approximation and prediction," *Journal of Intelligent and Fuzzy Systems*, vol. 24, no. 4, pp. 693–701, 2013.

[8] J. Du, Y. Yang, D. Wang, and C. Guo, "A robust adaptive neural networks controller for maritime dynamic positioning system," *Neurocomputing*, vol. 110, no. 13, pp. 128–136, 2013.

[9] H. G. Han, Z. L. Lin, and J. F. Qiao, "Modeling of nonlinear systems using the self-organizing fuzzy neural network with adaptive gradient algorithm," *Neurocomputing*, vol. 266, pp. 1447–1459, 2017.

[10] J. Dong, Y. Wang, and G. H. Yang, "Output feedback fuzzy controller design with local nonlinear feedback laws for discrete-time nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 40, no. 6, pp. 1447–1459, 2010.

[11] H. K. Lam and J. Lauber, "Membership function dependent stability analysis of fuzzy model based control systems using fuzzy Lyapunov functions," *Information Sciene*, vol. 232, pp. 253–266, 2013.

[12] J. Wang and J. Xiao, "Constructing fuzzy wavelet network modeling," in *Proceedings of the 2005 International Conference on Intelligent Computing*, pp. 169–172, Hefei, China, 2005.

[13] P. P. Preseren and B. Stopar, "Wavelet neural network employment for continuous GNSS or bit function construction: application for the assisted-GNSS principle," *Applied Soft Computing*, vol. 13, no. 5, pp. 2526–2536, 2013.

[14] Y. Bodyanskiy, A. Dolotov, and O. Vynokurova, "Evolving spiking wavelet neuro fuzzy self-learning system," *Applied Soft Computing*, vol. 14, no. 1, pp. 252–258, 2014.

[15] D. Bayram and S. Seker, "Wavelet based neuro detector for low frequencies of vibration signals in electric motors," *Applied Soft Computing*, vol. 13, no. 5, pp. 2683–2691, 2013.

[16] M. K. Shahriari, F. Sheikholeslam, and M. Zekri, "Design of adaptive fuzzy wavelet neural sliding mode controller for uncertain nonlinear systems," *ISA Transactions*, vol. 52, no. 3, pp. 342–350, 2013.

[17] R. Cheng, P. Yan, and Y. Bai, "A novel approach to fuzzy wavelet neural network modeling and optimization," *Electrical Power & Energy Systems*, vol. 64, pp. 671–679, 2015.

[18] A. Ebadat, N. Noroozi, A. A. Safavi, and S. M. Mousavi, "New fuzzy wavelet network for modeling and control: the modeling approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, pp. 3385–3396, 2011.

[19] M. Davanipoor, M. Zekri, and F. Sheikholeslam, "Fuzzy wavelet neural network with an accelerated hybrid learning algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 3, pp. 463–470, 2012.

[20] R. Chandra, "Adaptive problem decomposition in cooperative coevolution of recurrent networks for time series prediction," *Neurocomputing*, vol. 86, no. 4, pp. 1–8, 2013.

[21] R. M. Hou, W. Li, Q. Gao, Y. Hou, and C. Wang, "Indirect adaptive fuzzy wavelet neural network with self-recurrent consequent part for AC servo system," *ISA Transactions*, vol. 70, pp. 298–307, 2017.

[22] J. B. Oliveira, J. Moura, P. B. Oliveira, and H. Freire, "A swarm intelligence-based tuning method for the sliding mode generalized predictive control," *ISA Transactions*, vol. 53, pp. 1501–1515, 2014.

[23] S. Sahin and M. A. Cavuslu, "FPGA implementation of wavelet neural network training with PSO/iPSO," *Journal of Circuits, Systems, and Computers*, vol. 27, no. 6, pp. 15–30, 2018.

[24] X. G. Fu, S. H. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training recurrent neural networks with the Levenberg-Marquardt algorithm for optimal control of agrid-connected converter," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 6, no. 9, pp. 1900–1912, 2015.

[25] A. M. Rather, A. Agarwal, and V. N. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Systems with Applications*, vol. 42, pp. 3234–3241, 2015.

[26] L. Liu, Y. J. Liu, A. Chen, S. Tong, and C. L. P. Chen, "Intergal Barrier Lyapunov function-based adaptive control for switched nonlinear systems," *Science China Information Sciences*, vol. 63, pp. 132203–132211, 2020.

[27] B. Z. Xia, D. Cui, S. Zhen et al., "State of charge estimate of lithium-ion batteries using optimized Levenberg-Marquardt wavelet neural netwok," *Energy*, vol. 153, no. 15, pp. 694–705, 2018.

[28] H. G. Han, Y. N. Guo, and J. F. Qiao, "Nonlinear system modeling using a self-organizing recurrent radial basis function neural network," *Applied Soft Computing*, 2017.

[29] H. G. Han, Y. N. Guo, and J. F. Qiao, "Nonlinear system modeling using a self-organizing recurrent radial basis function neural network," *Applied Soft Computing*, vol. 71, pp. 1105–1116, 2018.