Sensors & Transducers, Vol. 171, Issue 5, May 2014, pp. 141-148

Sensors & Transducers

Application of a Self-recurrent Wavelet Neural Network in the Modeling and Control of an AC Servo System

*** Run Min HOU, Yuan Long HOU, Rong Zhong LIU, Guo Lai YANG, Qiang GAO**

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

Tel.: +86 25-84315417, fax: +86-25-84303262

E-mail: riluo1102@hotmail.com, yuanlongh@sohu.com, liurongz116@163.com, yyanggl@mail.njust.edu.cn, gaoqiang1007@yahoo.com.cn

Received: 23 April 2014 /Accepted: 27 May 2014 /Published: 31 May 2014

Abstract: To control the nonlinearity, widespread variations in loads and time varying characteristic of the high power ac servo system, the modeling and control techniques are studied here. A self-recurrent wavelet neural network (SRWNN) modeling scheme is proposed, which successfully addresses the issue of the traditional wavelet neural network easily falling into local optimum, and significantly improves the network approximation capability and convergence rate. The control scheme of a SRWNN based on fuzzy compensation is expected. Gradient information is provided in real time for the controller by using a SRWNN identifier, so as to ensure that the learning and adjusting function of the controller of the SRWNN operate well, and fuzzy compensation control is applied to improve rapidity and accuracy of the entire system. Then the Lyapunov function is utilized to judge the stability of the system. The experimental analysis and comparisons with other modeling and control methods, it is clearly shown that the validities of the proposed modeling scheme and control scheme are effective. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: AC servo system, Self-recurrent wavelet neural network, Fuzzy compensator, Lyapunov function, Gradient information.

1. Introduction

With the advancement of technology, high-power AC servo systems have become more and more widely utilized. For a servo driving system, the control system is required to have both a strong steady-stage and dynamic performance, and it is necessary to build a precise dynamic model of the system for conducting the analysis, simulation and control of an AC servo system. As a controlled object, the dynamic mathematical model of a high-power AC motor is a complex system, which is characterized by a heavy varying-load, slow time variation, nonlinearity and uncertain disturbance. Therefore, it is difficult to build a meticulous model by using traditional methods alone.

As neural networks may approach to any nonlinear function, they have already been widely utilized in the modeling of nonlinear systems. Some scholars [1-6] have proposed the building of motor servo system models by using neural networks, and the research results have shown that the precision of the neural network for the identification of nonlinear system was higher than that of the linear system.

However, when the sigmoid function was utilized as the excitation function of the back propagation (BP) neural system, the convergence rate of BP neural system slowed down. [7-10] proposed the identification and control algorithm by using the self-recurrent neural network information storage, but the training intensity of the related parameters was heavy, as the network structure of the self-recurrent neural network was rather complex [11]. As a feed forward network proposed on the basis of wavelet analysis, wavelet neural network effectively combines the structural model of a neural network with the multi-resolution and multi-scale analysis of the signals, and thus provides a theoretical basis for the determination of wavelet basis and the entire network structure, and avoids the blindness involved in structure design; the linear distribution of the network weight coefficients and the convexity of objective function enable the training process to fundamentally avoid local optimum and other nonlinear optimization problems [12, 13]. However, the gradient descent applied in wavelet neural network (WNN) training may cause the network to fall into local optimum, and the SRWNN has a strong feedback and direct control capability, as well as the on-line learning ability of neural networks and signal decomposition and identification ability of wavelets [14, 15].

In this study, in order to achieve control over the high-power AC servo system, a SRWNN was applied to build the intelligent modeling of this system, which was trained by means of self-adaptive learning rate. The convergence rate was greatly accelerated, and local optimum was avoided. With a satisfying model, gradient information was provided in real time for the controller by using the SRWNN identifier, which improved the learning efficiency of the WNN controller, and fuzzy compensation control was applied to improve the stability and rapidity of the whole system [16]. The simulation results show that the proposed method can be used to achieve the precise modeling and control of a high-power AC servo system.

2. Modeling AC Servo System

The control structure chart of an AC servo system is shown in Fig. 1. Due to the nonlinearity of the motor itself, the system load changes brought nonlinear phase comparison is very small, so the derivation of the model made the following assumptions [17, 18]:

1) No saturation effect;

2) Motor evenly distributed air gap, magnetic induction EMF sinusoidal shape;

3) Excluding the hysteresis and eddy current loss;

4) No rotor excitation winding;

Based on the above assumptions, the mathematics mode of permanent-magnet synchronous motor in the stationary (*d-q*) reference frame is shown as below:

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

where i_d, i_q; u_d, u_q and L_d, L_q are the currents, voltages and self inductance of the motor *d* axis and *q* axis respectively; R is the stator resistance of the motor(ohm); ψ_f is the permanent magnet flux of the motor; P is the pole-pairs of motor; J and B are the inertia constant of the motor and viscous friction coefficient; ω_r is the angular velocity of motor; T_e is the motor electromagnetic torque; T_L is the Load torque;

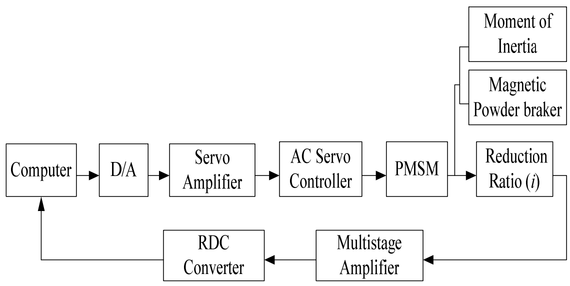

Fig. 1. The structure of the AC servo system.

The system is designed for the three closed-loop control system to accomplish the position of motor by the field oriented control technology, and then the high performance is obtained. Moreover, the method of the $i_d = 0$ vector control is utilized in order to simplify the motor control system. In the condition of $i_d = 0$, the mechanical equation of the motor can be shown as:

$$J\dot{\omega}_b + B\omega_b + T_L = T_e \tag{2}$$

where the ω_b is the mechanical angular velocity; and T_e can be expressed as:

$$T_e = \frac{3}{2} p\psi_f i_q = K_t i_q \tag{3}$$

where K_t is the torque constant.

Generally, the current time constant in the motor is much smaller than the mechanical time constant, so the delay time of the current response can be ignored. For the convenience of designing the controller, let variable $x_1 = \theta$ and $x_2 = \omega_r = \dot{\theta}$, put Eq. (2) into the Eq. (3), and then the state space equation of the AC servo system is:

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

where $-\frac{B}{J}$; $\frac{K_t}{J}$ and $-\frac{1}{J}T_L$ denote the nonlinear dynamic equations and the external disturbance.

3. Self-recurrent Wavelet Neural Network

3.1. SRWNN Structural Description

The mother wavelet layer of the SRWNN possesses self-feedback neurons, which are able to obtain information of the network in the past state. As a result, it can quickly adapt to the mutation of the control environment and eliminate the oscillation characteristics of the general wavelet neural network. The structure of the SRWNN is shown in Fig. 2, which includes N_i inputs, one output and $N_w \times N_i$ mother wavelets. The network has a four-layer network structure [14]:

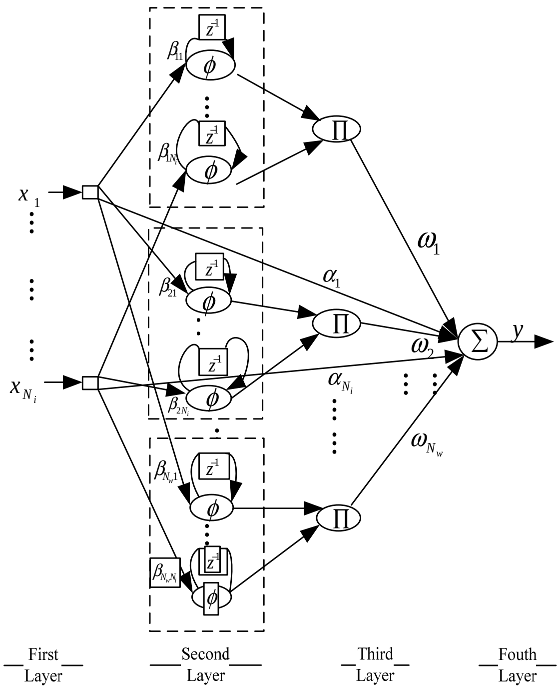

Fig. 2. Structural chart of SRWNN.

1) The first layer is the input layer, which directly transmits the input variable to the next layer.

2) The second layer is the mother wavelet layer, where each node includes a mother wavelet and a self feedback loop. We chose the Gaussian function $\phi(x) = -x\exp(-\frac{1}{2}x^2)$ as the mother wavelet function, and the wavelet of each node ϕ_{jk} could be transferred from mother wavelet directly, as follows:

$$\phi_{jk}(z_{j,k}) = \phi_{jk}\frac{u_{jk} - m_{jk}}{d_{jk}} \quad (5)$$

where $z_{jk} = \frac{u_{jk} - m_{jk}}{d_{jk}}$, u_{jk} represents the output, m_{jk} represents the shift factor, d_{jk} represents the spreading factor, and j, k respectively represent the j^{th} wavelet of the k^{th} input item. Then, the input at n moment was determined as follows:

$$u_{jk} = x_k(n) + \beta_{jk}\phi_{jk}(n-1) \quad (6)$$

where β_{jk} is the self feedback information storage ratio, and $\phi_{jk}(n-1)$ shows that the algorithm contained the memory in the past state.

3) The third layer is the product layer, which could be expressed as follows:

$$\phi_j = \prod_{k=1}^{N_i}\phi_{jk}(z_{jk}) \quad (7)$$

4) The fourth layer is the output layer, which included the first and third inputs, and could be expressed as follows:

$$y = \sum_{j=1}^{N_w}\omega_j\phi_j(x) + \sum_{k=1}^{N_i}\alpha_k x_k \quad (8)$$

where ω_j is the weight from the output ϕ_j of the third to the fourth layer, and α_k is the weight of the first output transmitted to the fourth layer.

3.2. Training Algorithm

The self-adaptive learning rate which obtained the time varying learning rate instead of innate gradient descent was adopted, so as to acquire a fast convergence and avoid local optimum and other problems. It was assumed that the performance function of SRWNN learning algorithm was as follows:

$$J = \frac{1}{2}e^2(k) = \frac{1}{2}(y_{out}(k) - y(k)^2), \quad (9)$$

where y_{out} is the target output. By using the gradient descent training method, the modification of weight could be expressed as follows:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \Delta\mathbf{W}(n) - \\ - \eta(n)(\frac{\partial J}{\partial \mathbf{W}}), \quad (10)$$

where $\boldsymbol{\eta} = \text{diag}\{\eta_\alpha\ \eta_m\ \eta_d\ \eta_\beta\ \eta_\omega\}$ is the self-adaptive learning rate of the weight, and the weight vector $\mathbf{W} = [\alpha_k, m_{jk}, d_{jk}, \beta_{jk}, \omega_j]$ respectively. The derivative of performance function with respect to the weight is $\frac{\partial J}{\partial \mathbf{W}} = -e\frac{\partial y}{\partial \mathbf{W}}$. By using the chain recurrence method and the partial derivatives of the outputs with respect to the weight were,

$$\frac{\partial y}{\partial m_{jk}} = -\frac{\omega_j}{d_{jk}}\frac{\partial \phi_j}{\partial z_{jk}}, \frac{\partial y}{\partial a_k} = x_k, \frac{\partial y}{\partial d_{jk}} = -\frac{\omega_j}{d_{jk}}\frac{z_{jk}\partial \phi_j}{\partial z_{jk}},$$

$$\frac{\partial y}{\partial \beta_{jk}} = -\frac{\omega_j}{d_{jk}}\frac{\phi_{jk}\partial \phi_j}{\partial z_{jk}};$$

The $\frac{\partial \phi_j}{\partial z_{jk}} = (z_{jk}^2 - 1)\exp(-z_{jk}^2/2)$; $\frac{\partial y}{\partial \omega_j} = \phi_j(x)$ and the self-adaptive learning rate $\eta_i(k)$ was determined as follows [19]:

$$\eta_i(k) = \begin{cases} \dfrac{\eta_{o,i}}{\left(\left\|\frac{\partial y(k)}{\partial W_1}\right\|^2 + \left\|\frac{\partial y(k)}{\partial W_2}\right\|^2 + \left\|\frac{\partial y(k)}{\partial W_3}\right\|^2 + \left\|\frac{\partial y(k)}{\partial W_4}\right\|^2\right)} & i = 1,5 \\ \dfrac{\eta_{o,i}}{\left\|\frac{\partial y(k)}{\partial W_i}\right\|^2} & i = 2,3,4 \end{cases} \quad (11)$$

where $0 < \eta_{0,i} < 2$.

4. Servo System Controller

In order to obtain a stronger control performance, we replaced the nonlinear control object with the SRWNN identifier; then back-propagated the gradient information required by the weight modification to the controller of the SRWNN; and applied fuzzy compensator to improve the rapidity of the control system. Fig. 3 shows the SRWNN structural schematic with fuzzy compensation.

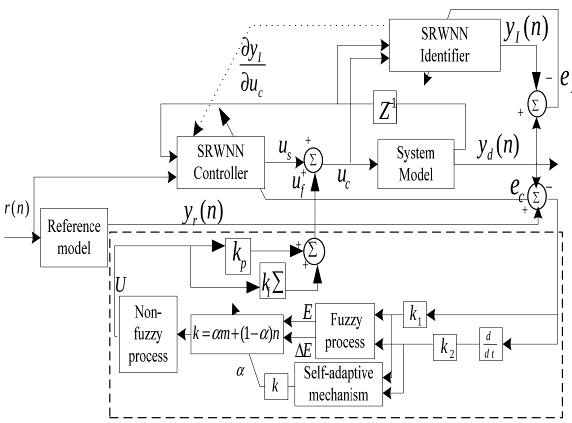

Fig. 3. Structural of fuzzy compensation SRWNN.

It can be seen from Fig. 3 that the difference e_I between the identifier output y_I and actual output y_d was used to train the SRWNN identifier; and the control output signal u_c of the system was as follows:

$$u_c = u_s + u_f, \quad (12)$$

where u_s is the control output of the SRWNN. The task of the controller was to determine the control signal u_c and ensure the control error e_c between the output y_d and output of the reference model y_r is the smallest with the same reference input u_f is the control output of the fuzzy control compensator. It was assumed that the performance function of the controller of SRWNN u_s was as follows:

$$J_s = \frac{1}{2}(y_r(k) - y_d(k)^2) = \frac{1}{2}e_c^{\,2}(k) \quad (13)$$

The weight $\mathbf{W}_s$ of the controller of SRWNN was as follows:

$$\Delta \mathbf{W}_s(n) = -e_c\frac{\partial y_d}{\partial u_c}\frac{\partial u_c}{\partial \mathbf{W}_s} \quad (14)$$

Due to the fact that the error between the network identification output y_I and actual output of the controlled object y_d was within the range of accuracy requirement of the training, we considered replacing y_d approximately with y_I, which may not greatly affect the control performance. Then, the controlled object gradient information $\frac{\partial y_d}{\partial u_c}$ of the learning algorithm of the SRWNN controller was replaced by $\frac{\partial y_I}{\partial u_c}$. $\frac{\partial u_c}{\partial \mathbf{W}_s}$ could be calculated by using the training algorithm of SRWNN.

Given the direct impact of the performance of the compensating controller on the compensation effect of the system over the nonlinear characteristics, the design of the compensating controller was the key of the system design. Owing to the fact that the performance of the controller depended mainly on the fuzzy control rule, we used a fuzzy controller which could automatically adjust the control rule to improve the performance of the entire system [20]. The control principle chart had proven to be within the range of the dotted line in Fig. 3.

In Fig. 3, k_1, k_2 are the scale factor of control error and control error differential, respectively; α is a self-adaptive factor which can adjust rules, and the variable is related to E and ΔE, i.e.:

$$\alpha(t)=\alpha(t-1)-\eta\varepsilon(E\Delta E)\,, \tag{15}$$

where η is the change rate of adjusting α, and variable ε is determined as follows:

$$\varepsilon(l)=\begin{cases}1 & l>1\\ l & -1\le l\le 1\,,\\ -1 & l<-1\end{cases} \tag{16}$$

where m, n, k are respectively the m-, n-, k-th fuzzy set of fuzzy control rule $E_m, \Delta E_n$ and U_k; k can be expressed as shown in Reference [21]:

$$k=<\alpha m+(1-\alpha)n>\,, \tag{17}$$

where $<\cdot>$ is the rounding symbol.

The universe of E, ΔE and U are respectively $[X, X]$, $[Y, Y]$ and $[Z, Z]$, and the membership function corresponding to the fuzzy variable is shown in Fig. 4.

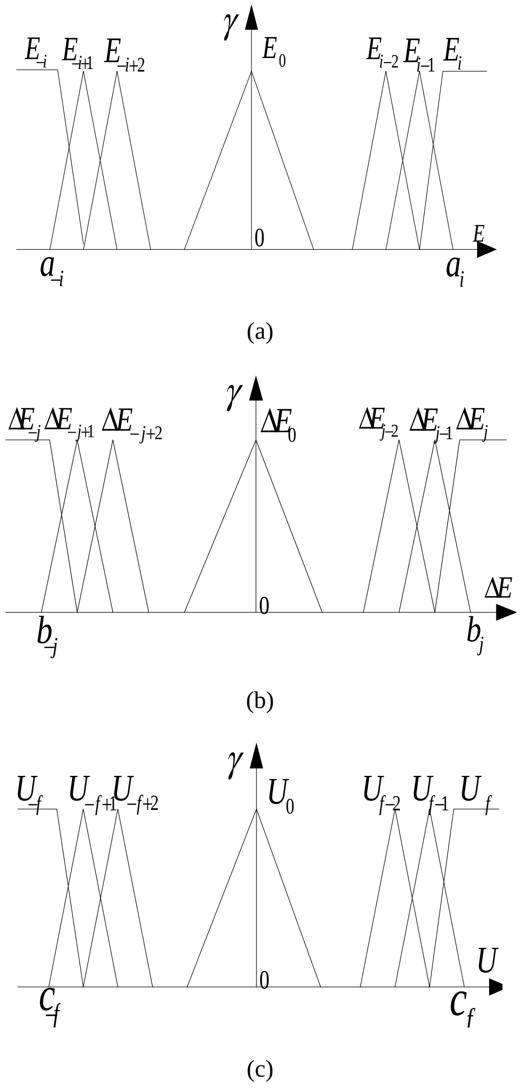

Fig. 4. Membership function.

It can be seen from Fig. 4 that the interval of the fuzzy subset U_k was $S=\frac{Z}{f}$. In order to avoid the loss of generality, we considered that when $a_m \le E \le a_{m+1}$ and $b_n \le \Delta E \le b_{n+1}$, there were four control rules, and the output of fuzzy controller U was as follows:

$$U=\frac{B}{\sum_{p=1}^{4} W_p} \tag{18}$$

where

$$\begin{aligned}B=\{&w_1[am+(1-a)n]+\\ &w_2[a(m+1)+(1-a)n]+\\ &w_3[am+(1-a)(n+1)]\\ &+w_4[a(m+1)+(1-a)(n+1)]\}S\end{aligned} \tag{19}$$

$$\sum_{p=1}^{4} W_p=\begin{cases}w_1=\min[\gamma_{E_m}(E),\ \ \gamma_{\Delta E_n}(\Delta E)]; & p=1\\ w_2=\min[\gamma_{E_{m+1}}(E),\ \gamma_{\Delta E_n}(\Delta E)]; & p=2\\ w_3=\min[\gamma_{E_m}(E),\ \gamma_{\Delta E_{n+1}}(\Delta E)]; & p=3\\ w_4=\min[\gamma_{E_{m+1}}(E), \gamma_{\Delta E_{n+1}}(\Delta E)]; & p=4\end{cases} \tag{20}$$

When the output U is derived by the fuzzy control algorithm, it is necessary to transform the domain of discourse by multiplying the variable with the corresponding scale factor. k_p, k_i are the proportionality factor and integral coefficient, respectively, thus the actual output of the fuzzy compensator u_f is as follows:

$$u_f(\mathrm{k}+1)=k_p U(\mathrm{k}+1)+k_i\sum_{i=0}^{k} U(i) \tag{21}$$

In the formula, $k_p U$ ensured that the system could respond rapidly in the transient state, when in the steady state, there was no static error or oscillation in the system response, due to the integral action of $k_i \sum U$. As a consequence, the speed of the system was improved.

5. Analysis Stability of SRWNN

The Lyapunov function was adjusted to judge the stability of the system. It was assumed that the corresponding Lyapunov discrete function of SRWNN was as follows:

$$V_c(n)=\frac{1}{2}(y_r(n)-y_d(n)^2)=\frac{1}{2}e_c^{\,2}(n)\,, \tag{22}$$

Theorem 1. By assuming the variable value $\boldsymbol{\eta}$ = diag$\{\eta_\alpha, \eta_m, \eta_d, \eta_\beta, \eta_\omega\}$ = diag$\{\eta_1, \eta_2, \eta_3, \eta_4, \eta_5\}$

was the weight learning rate of SRWNN, $\boldsymbol{C}_{\max}$ was defined as follows:

$$\begin{aligned}\boldsymbol{C}_{\max} &= [C_{1\max}\ C_{2\max}\ C_{3\max}\ C_{4\max}\ C_{5\max}]^{\mathrm{T}} \\ &= [\max_n \left\| \frac{\partial u_c}{\partial \alpha} \right\|\ \max_n \left\| \frac{\partial u_c}{\partial m} \right\|\ \max_n \\ &\left\| \frac{\partial u_c}{\partial d} \right\|\ \max_n \left\| \frac{\partial u_c}{\partial \beta} \right\|\ \max_n \left\| \frac{\partial u_c}{\partial \omega} \right\|]\end{aligned} \quad (23)$$

If η_i could satisfy $0 < \eta_i < \frac{2}{(\mu C_{i\max}^2)}$, then SRWNN was convergent. $i = 1, 2, ..5$, μ was defined as a positive constant.

The proof as follows:

$$\begin{aligned}\Delta V_c(n) &= \frac{1}{2}[e_c^{\,2}(n+1) - e_c^{\,2}(n)] \\ &= \Delta e_c(n)[e_c(n) + \frac{1}{2}\Delta e_c(n)] \\ &= \eta_i [\frac{\partial e(n)}{\partial \omega_i(n)}]^{\mathrm{T}} e(n) [\frac{\partial u_c(n)}{\partial \omega_i(n)}] \\ &(e(n) + \frac{1}{2}[\frac{\partial e(n)}{\partial \omega_i(n)}]^{\mathrm{T}} e(n) [\frac{\partial u_c(n)}{\partial \omega_i(n)}]) \\ &= -\gamma e^2{}_c(n),\end{aligned} \quad (24)$$

where $\gamma \geq \mu \eta_i C_{i\max}^2 (1 - \frac{1}{2}\mu \eta_i C_{i\max}^2)$.

It can be seen from Eq. (24) that if $0 < \eta_i < \frac{2}{(\mu C_{i\max}^2)}$.

Then $\mu \eta_i C_{i\max}^2 (1 - \frac{1}{2}\mu \eta_i C_{i\max}^2) > 0$; Thus, the SRWNN was convergent.

6. Simulation Experiment and Analysis

6.1. Modeling and Simulation

Put the multi-amplitude signal of pseudorandom into the experiment table of the AC servo system with a rated power of 9.7 KW, and received a set of output data, as shown in Fig. 5. To improve the algorithm learning efficiency and accelerate the convergence rate, we conducted normalization processing on the input/output data of the AC servo system. The input signal is the control voltage signal from -5 V to 5 V, as shown in Fig. 5(a), and output signal is as shown in Fig. 5(b). The sampling period was 10 ms, and there were 1000 sets of data, among which the former 600 sets were utilized for training, and the latter 400 sets were used for testing.

The main parameters in the high power AC system were as follows system load rotational inertia: $J = 8200 kg \cdot m^2$; the friction moment of the load: $850 kg \cdot m^2$; the system load disturbing moment: $93200 kg \cdot m^2$; and the reduction ratio: $i = 1039$. The structure set of SRWNN was 4-5-5-1, i.e. the first layer was the input layer with four inputs; the second layer was the mother wavelet layer with five mother wavelets in each node; the third layer was product layer with five mother wavelet groups; and the fourth layer was the output layer with one output. For comparison, it can be provided the identification consequences of the WNN. Two performance indexes were applied to measure the two modeling methods; one was the root mean squared error (RMS) between the modeling output and actual system output, and the other was the variance accounted for (VAF) showing the similarity degree between the two signals. The smaller the RMS was and the larger the VAF was, the more effective the model was and the closer it was to the actual system. The WNN and SRWNN identification model output and desired output are illustrated in Fig. 6 and Fig. 7.

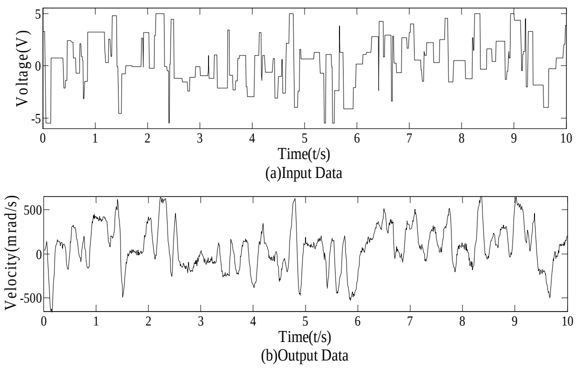

Fig. 5. Input/output data.

Fig. 6(a) and Fig. 7(a) show the training data of the WNN and SRWNN; Fig. 6(b) and Fig. 7(b) show the validating data of the WNN and SRWNN; and the black line and red line are the network output and actual output, respectively. Table 1 displays the RMS and VAF of the transfer function output and the actual output.

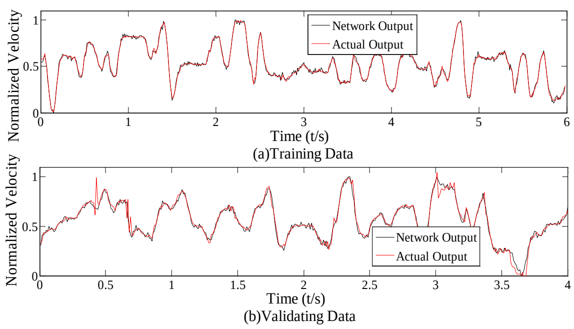

Fig. 6. Identification model output and desired output of WNN.

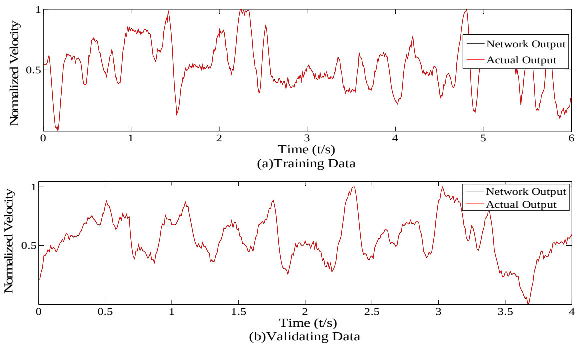

Fig. 7. Identification model output and desired output of SRWNN.

Table 1. Performance of two AC servo system models.

	Training RMS	Training VAF	Validating RMS	Validating VAF
SRWNN	0.00216	99.816 %	0.00257	99.895 %
WNN	0.0231	98.963 %	0.0416	95.128 %

It can be viewed from the comparison in Table 1 that the SRWNN identification improves the identification precision of the traditional WNN. It can also effectively solve the problem of the network easily falling into local optimum, due to its strong generalization ability, and it can well describe the dynamic characteristics of the high power AC servo system

6.2. Performance of Control System

In order to test and verify the effectiveness of the fuzzy compensation controller in SRWNN, the SRWNN controller is been used to compare it. Fig. 8 shows the position response curve added with a $300 N \cdot m$ step disturbance at 1.8 s. It can be seen from Fig. 8 that when the load is disturbed, there is a significant deviation of position response by the SRWNN control algorithm, and the position response requires a long period of time to return to the stable position. However, when applying the fuzzy compensation controller algorithm in SRWNN, the system has rather strong robustness it can achieve the goal position in a short period of time.

The step response of the fuzzy compensation control in SRWNN is utilized. Fig. 9 and Fig. 10 are respectively the system response of SRWNN control and fuzzy compensation control of SRWNN with sine-input signal. It can be seen from the figures that the fuzzy compensation SRWNN can track the sinusoidal signal closely, and that the maximum error is less than 2*mrad*; in addition, the SRWNN has a large error when tracking the sinusoidal signal, and the maximum error is larger than 8*mrad*. Therefore, the fuzzy compensation SRWNN has a better dynamic performance and steady state performance than the SRWNN control, and it can effectively increase the stability and speed of the system.

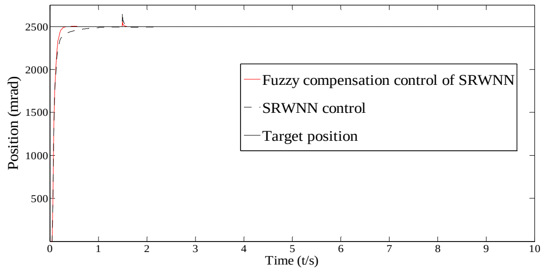

Fig. 8. Step response curve of load disturbance.

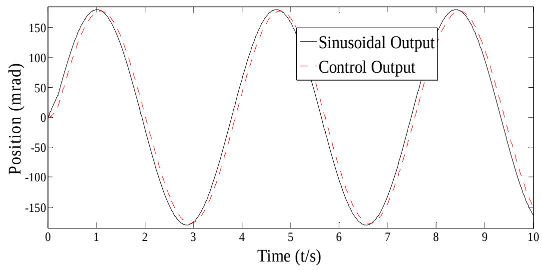

Fig. 9. Sinusoidal response curve of SRWNN control.

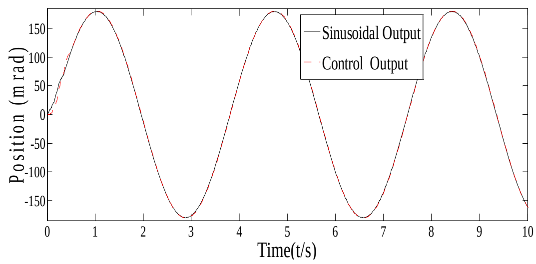

Fig. 10. Sinusoidal response curve of fuzzy compensation control in SRWNN.

7. Conclusions

In this paper, aiming at the modeling and control problems of a typical nonlinear system as a high-power AC servo system, SRWNN was applied to conduct system modeling; the training of the network was achieved by the self-adaptive learning rate algorithm; and based on the precise system model, the fuzzy compensation controller in SRWNN was further designed for system control. The verification and comparison of the simulation results and other modeling and control methods showed that this algorithm successfully solved the problem of the WNN easily falling into local optimum; the algorithm showed a strong generalization ability and high precision of identification model; and the stability and rapidity of the entire system trace was improved by adding fuzzy compensation in the control method.

Acknowledgements

This work was supported by the Research program of the Ministry of Science and Technology

(MoST) of China through the National Program on Key Basic Research Project of China under Grant No. 61311603.

References

[1]. Z. D. Qi, X. J. Zhu and G. Y. Cao, Neural networks modeling of MCFC system and fuzzy control research based on FPGA, in *Proceedings of the 5th IEEE World Congress on Intelligent Control and Automation, (IEEE'04)*, Hangzhou, China, 2004, pp. 2486-2490.

[2]. Z. D. Qi, X. J. Zhu and G. Y. Cao, Voltage model of DMFC based on neural networks identification, *Computer Simulation*, Vol. 22, Issue 5, 2005, pp. 92-94.

[3]. Q. Gao, Y. Jin, Y. L. Hou and L. J. Ji, Modeling and control for mine sweeping plough electro-hydraulic servo system of certain mine-clearing vehicle, *Journal of Nanjing University of Science and Technology*, Vol. 36, Issue 2, 2012, pp. 238-244.

[4]. S. N. Huang, K. K. Tan and T. H. Lee, Nonlinear adaptive control of interconnected systems using neural networks, *IEEE Transactions on Neural Networks*, Vol. 17, Issue 1, 2006, pp. 243-246.

[5]. T. Chen and H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Transactions on Neural Networks*, Vol. 6, Issue 4, 1995, pp. 911-917.

[6]. J. I. Canelon, L. S. Shieh and N. B. Karayiannis, A new approach for neural control of nonlinear discrete dynamic systems, *Information Science*, Vol. 1, Issue 74, 2005, pp. 177-196.

[7]. C. H. Lee and C. C. Teng, Identification and control of dynamic systems using recurrent fuzzy neural network, *Fuzzy Systems*, Vol. 8, No. 4, 2000, pp. 349-366.

[8]. F. J. Lin, H. J. Shieh, L. T. Teng and P. H. Shieh, Hybrid controller with recurrent neural network for magnetic levitation system, *Magnetics*, Vol. 41, Issue 7, 2005, pp. 2260-2269.

[9]. X. Li and W. Yu, Dynamic system identification via recurrent multilayer perceptrons, *Information Science*, Vol. 1, Issue 47, 2002, pp. 45-63.

[10]. F. J. Lin, H. J. Shieh, P. H. Shieh and P. H. Shen, An adaptive recurrent-neural-network motion controller for X-Y table in CNC machine, *IEEE Transactions on System, Man and Cybernetics*, B, Vol. 36, Issue 2, 2006, pp. 286-299.

[11]. T. W. S. Chow and Y. Fang, A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics, *IEEE Transactions on Industrial Electronics*, Vol. 45, Issue 1, 1998, pp. 151-161.

[12]. Q. H. Zhang and A. Benveniste, Wavelet neural, *IEEE Transactions on Neural Networks*, Vol. 3, Issue 6, 1992, pp. 889-989.

[13]. S. J. Yoo, J. B. Park and Y. H. Choi, Stable predictive control of chaotic systems using self-recurrent wavelet neural network, *International Journal of Control, Automation, and Systems*, Vol. 3, Issue 1, 2005, pp. 316-321.

[14]. S. J. Yoo, J. B. Park and Y. H. Choi, Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates, *Information Sciences*, Vol. 177, Issue 15, 2007, pp. 3074-3098.

[15]. S. J. Yoo, J. B. Park and Y. H. Choi, Stable predictive control of chaotic systems using self-recurrent wavelet neural network, *International Journal of Control* Automated Systems, Vol. 3, Issue 1, 2005, pp. 43-55.

[16]. K. Qian, S. S. Xie, W. Zhang and D. J. Yu, Application of speed control system based on fuzzy neural network compensation, *Electric Machines and Control*, Vol. 8, Issue 4, 2004, pp. 373-376.

[17]. Y. C. Zhu, D. W. Ma, Z. G. Li and G. G. Le, Optimal sliding mode servo control with integral term for rocket launcher, *Acta Armamentarii*, Vol. 28, Issue 10, 2007, pp. 1272-1275.

[18]. Q. Gao, Z. Sun, G. L. Yang, R. M. Hou, L. Wang and Y. L. Hou, A novel active disturbance rejection-based control strategy for a gun control system, *Journal of Mechanical Science and Technology*, Vol. 26, Issue 12, 2012, pp. 4141-4148.

[19]. H. Yang, Y. F. Gao, F. Luo, and Y. G. Xu, One-step-ahead predictive control using adaptive self-recurrent wavelet neural networks, *Information Control*, Vol. 39, Issue 5, 2010, pp. 553-558.

[20]. J. W. Li, K. D. Zhao and S. L. Wu, Application of adaptive control in hydraulic simulator based on fuzzy compensator, *Acta Aeronautica et Astronautica Sinica*, Vol. 24, Issue 1, 2003, pp. 72-74.

[21]. W. Z. Qiao, W. P. Zhuang, T. H. Heng and S. S. Shan, A rule self-regulating fuzzy controller, *Fuzzy Sets and Systems*, Vol. 47, Issue 1, 1992, pp. 13-21.

___