Self-Constructing Fuzzy Neural Network Speed Controller for Permanent-Magnet Synchronous Motor Drive

Faa-Jeng Lin, *Senior Member, IEEE*, Chih-Hong Lin, and Po-Hung Shen

***Abstract*—A self-constructing fuzzy neural network (SCFNN) which is suitable for practical implementation is proposed in this study. The structure and the parameter learning phases are performed concurrently and online in the SCFNN. The structure learning is based on the partition of input space and the parameter learning is based on the supervised gradient decent method using a delta adaptation law. Several simulation and experimental results are provided to demonstrate the effectiveness of the proposed SCFNN control stratagem with the implementation of a permanent-magnet synchronous motor (PMSM) speed drive. Moreover, the simulation results of time varying and nonlinear disturbances are given to show the dynamic characteristics of the proposed controller over a broad range of operating conditions.**

***Index Terms*—Fuzzy neural network, gradient decent method, self-constructing, synchronous motor.**

I. Introduction

Fuzzy neural networks (FNNs) combine the capability of fuzzy reasoning in handling uncertain information [1]–[3] and the capability of neural networks in learning from processes [4]–[6]. Therefore, recently much research has been conducted by applying the fuzzy neural network systems in the control fields to deal with nonlinearities and uncertainties of the control systems [7]–[11]. Most traditional FNN applications are only parameter learning based on supervised backpropagation algorithm in which the parameters of the membership functions and the connected weights are adjusted, and the structure of the FNN has been determined and fixed in advance. Though the control performance of the fixed structure FNNs with the ability of online parameter learning are usually acceptable [10], [11], large numbers of nodes are usually needed in the hidden layers for complex controlled plant. On the other hand, several approaches consist of structure and parameter learning phases have been proposed in [12]–[15]. These two-phase learning algorithms not only extract the fuzzy logic rule from input data and adjust the fuzzy partitions of the input and output spaces but also adjust the parameters of the FNN. Traditionally, these two phases are done sequentially [13]; that is, the structure learning phase is employed to decide the structure of fuzzy rules first, then the parameter learning phase is used to tune the coefficients of each rule (for example, membership functions). The disadvantages of this sequential learning scheme are that it is suitable only for offline instead of online operation [13] and a large amount of representative data must be collected in advance for the implementation of this scheme. Moreover, the independent realization of the structure and parameter learning is too time consuming. To overcome these problems and to achieve accelerated learning, a self-constructing neural fuzzy inference network (SONFIN) was proposed in [15] to perform the structure and parameter learning phases concurrently. However, the network structure and parameter learning algorithms proposed in [15] are both complicated and not suitable for practical implementation. To overcome this difficulty, a self-constructing fuzzy neural network (SCFNN) is proposed in this study. The structure and the parameter learning phases are done concurrently and online in the SCFNN. In addition, the structure learning algorithm of SCFNN, which is based on the partition of the input space, is simpler than SONFIN and the parameter learning algorithm of SCFNN is based on the supervised gradient decent method [7]–[11] using the delta adaptation law proposed in [16]. Finally, a permanent-magnet synchronous motor (PMSM) speed drive system considering time-varying parameter variation and nonlinear friction force is implemented in this study to verify the effectiveness and robustness of the proposed SCFNN.

Manuscript received March 19, 2001; revised June 11, 2001. This work was supported by the National Science Council of Taiwan, R.O.C. under Grant NSC 89-2213-E-033-048.

F.-J. Lin and P.-H. Shen are with the Department of Electrical Engineering, National Dong Hwa University, Hualien 974, Taiwan.

C.-H. Lin is with the Department of Electrical Engineering, National Lien-Ho Institute of Technology, Miao Li 360, Taiwan.

Publisher Item Identifier S 1063-6706(01)09625-4.

II. PMSM Drive System

A block diagram of a SCFNN controlled PMSM speed drive system is shown in Fig. 1, where $\omega_r^*, \omega_r, \omega_m, e, \Delta e$ and e_m are the rotor speed command, rotor speed, output of the reference model, error between ω_r^* and ω_r, error change, and error between ω_m and ω_r, respectively. The PMSM drive consists of a PMSM loaded with a DC generator, a ramp comparison current-controlled pulsewidth modulation (PWM) voltage source inverter, a field-oriented mechanism. The PMSM used in this drive system is a three-phase four-pole 750 W 3.47 A 3000 rpm type. By using the field-oriented technique, the PMSM drive can be reasonably represented by the following equations [11]:

$$T_e = K_t i_{qs}^* \tag{1}$$

$$H_p(s) = \frac{1}{Js+B} = \frac{b}{s+a} \tag{2}$$

where

- T_e electromagnetic torque;
- K_t torque constant;
- i_{qs}^* q-axis current command in synchronous rotating reference frame [11];
- s Laplace operator;
- J inertia constant;
- B damping coefficient.

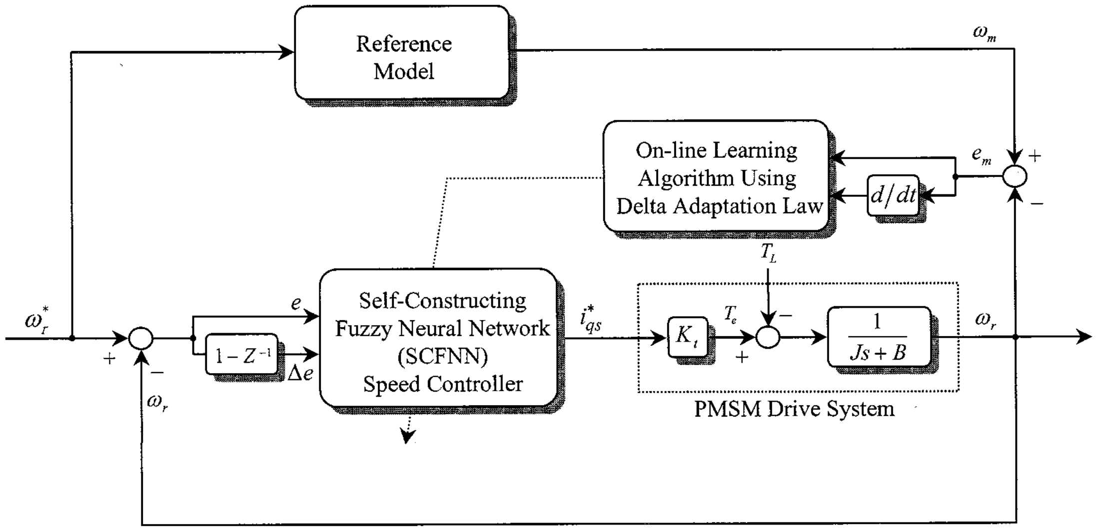

Fig. 1. Block diagram of SCFNN speed control system.

Detailed parameters of the drive system are (on a scale of 50 (rad/s)/V)

$$
\begin{aligned}
K_t &= 0.6732 \text{ Nm/A}, \quad a = 4.4, \quad b = 15.2\\
\bar{J} &= 1.32 \times 10^{-3} \text{ Nms}^2 = 0.066 \text{ Nmsrad/V}\\
\bar{B} &= 5.78 \times 10^{-3} \text{ Nms/rad} = 0.289 \text{ Nm/V}
\end{aligned}
$$

The overscore "‾" symbol represents the nominal value of the system parameter. The SCFNN speed controller and the on-line learning algorithm using the delta adaptation law which are shown in Fig. 1 will be introduced in the following sections.

III. Self-Constructing Fuzzy Neural Network

A. Description of Self-Constructing Fuzzy Neural Network

A fuzzy logic rule with constant consequence; part of the following form is adopted in the SCFNN:

$$R_j\text{: IF } x_1 \text{ is } A_1^j \text{ and } \cdots \text{ and } x_n \text{ is } A_n^j, \text{ THEN } y = b_j \tag{3}$$

where

- x_i y input and output variables, respectively;
- A_i^j linguistic term of the precondition part with membership function $u_{A_i^j}$;
- b_j constant consequent part;
- n number of input variables.

The structure of the SCFNN is shown in Fig. 2 where the functions of the node in each layer are described as follows:

Layer 1: Each node in this layer is an input node, which corresponds to one input variable. These nodes only pass the input signal to the next layer. In this study, the input variables are $x_1 = e$ (the speed error) and $x_2 = \Delta e$ (the derivative of speed error).

Layer 2: Each node in this layer acts as a linguistic label of one of the input variables in layer 1, i.e., the membership value specified the degree to which an input value belongs to a fuzzy

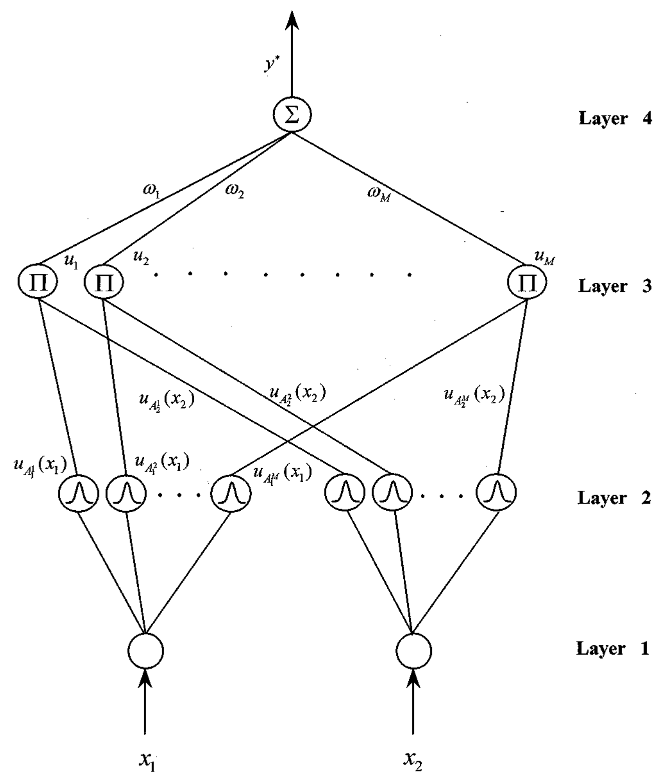

Fig. 2. Schematic diagram of SCFNN.

set is determined in this layer. The Gaussian function as follows is adopted as the membership function

$$u_{A_i^j} = \exp\left(-\frac{(x_i - m_{ji})^2}{\sigma_{ji}^2}\right) \tag{4}$$

where m_{ji} and σ_{ji} are the mean and standard deviation, respectively, of Gaussian functions of jth term associated with ith input variable x_i.

Layer 3: Each node in this layer represents the precondition part of one fuzzy logic rule. Therefore, each node of this layer is denoted by Π, which multiplies the incoming signals and outputs the product result, i.e., the firing strength of a rule. For the jth rule node

$$u_j = u_{A_1^j}(x_1)u_{A_2^j}(x_2)\ldots u_{A_n^j}(x_n) = \prod_i u_{A_i^j}(x_i) \quad (5)$$

where u_j is the output of the jth rule node.

Layer 4: This layer acts a defuzzifier. The single node in this layer is labeled Σ and it sums all incoming signals to obtain the final inferred result

$$y^* = \sum_{j=1}^{M} w_j u_j \quad (6)$$

where the link weight w_j is the output action strength associated with the jth rule and y^* is the output of the SCFNN. The consequent part of one fuzzy logic rule is implicitly contained in w_j. In this study, the output is $y^* = i_{qs}^*$.

B. Online Learning Algorithms for SCFNN

Two types of learning algorithms, the structure learning and the parameter learning, are developed to construct the SCFNN. The structure learning is used to find proper input space fuzzy partitions and fuzzy logic rules subject to two objectives: (1) to minimize the number of rules generated and (2) to minimize the number of fuzzy sets on the universe of discourse of each input variable. The parameter learning is based on supervised learning algorithms, the link weights in the consequent part and the parameters of membership functions are adjusted by the backpropagation algorithm to minimize a given energy function. Initially, there are only input and output nodes in the SCFNN without any membership and rule nodes. The membership and the rule nodes are generated automatically and dynamically in the learning process according to the online incoming data by performing the structure and parameter learning processes. The structure and parameter learning algorithms are described in the rest of this section in detail.

1) Structure Learning Phase: The first step in the structure learning is to determine whether or not to perform the structure learning. If $e_{\min} \le |e|$ or $\Delta e_{\min} \le |\Delta e|$, where $e_{\min}$ and $\Delta e_{\min}$ are preset positive constants, then the structure learning is necessary. Next, it will further decide whether or not to add a new node (membership function) in layer 2 and the associated fuzzy logic rule in layer 3. Since one cluster formed in the input space corresponds to one potential fuzzy logic rule, the firing strength of a rule for each incoming data x_i can be represented as the degree that the incoming data belong to the cluster. The firing strength obtained from (5) is used as the degree measure

$$D_j = u_j \quad j = 1, \ldots, Q(t) \quad (7)$$

where $Q(t)$ is the number of existing rules at time t. According to the degree measure, the criterion of generating a new fuzzy rule for new incoming data is described as follows. Find the maximum degree $D_{\max}$

$$D_{\max} = \max_{1 \le j \le Q(t)} D_j. \quad (8)$$

If $D_{\max} \le \bar{D}$, then a new membership function is generated where $\bar{D} \in (0,1)$ is a prespecified threshold, that should be decayed during the learning process to limit the size of SCFNN. Next the mean and standard deviation of the new membership function are assigned with prespecified values using heuristic or prior knowledge. Thus, the mean and standard deviation of the new membership function are selected as follows:

$$m_i^{(\text{new})} = x_i \quad (9)$$
$$\sigma_i^{(\text{new})} = \sigma_i \quad (10)$$

where x_i is the new incoming data and σ_i is a prespecified constant.

To avoid the newly generated membership function being too similar to the existing one, the similarities between the new membership function and the exiting ones must be checked. The similarity measure proposed in [13], [15] is used to check the similarity of two membership functions. Suppose there are two fuzzy sets A and B with membership functions $u_A(x) = \exp[-(x-m_1)^2/\sigma_1^2]$ and $u_B(x) = \exp[-(x-m_2)^2/\sigma_2^2]$, respectively, to be measured. Assume $m_1 \ge m_2$. It follows that $|A \cap B|$ can be computed by [13], [15]

$$\begin{aligned}|A \cap B| = {} & \frac{1}{2}\frac{h^2(x)[m_2 - m_1 + \sqrt{\pi}(\sigma_1+\sigma_2)]}{\sqrt{\pi}(\sigma_1+\sigma_2)} \\ & + \frac{1}{2}\frac{h^2(x)[m_2 - m_1 + \sqrt{\pi}(\sigma_1-\sigma_2)]}{\sqrt{\pi}(\sigma_1+\sigma_2)} \\ & + \frac{1}{2}\frac{h^2(x)[m_2 - m_1 + \sqrt{\pi}(\sigma_1-\sigma_2)]}{\sqrt{\pi}(\sigma_1-\sigma_2)} \quad (11)\end{aligned}$$

where $h(x) = \max\{0, x\}$. Then a suitable similarity measure [13], [15] is given using the entropy $E(A,B)$

$$E(A,B) \equiv \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{\sigma_1\sqrt{\pi} + \sigma_2\sqrt{\pi} - |A \cap B|}. \quad (12)$$

The similarity checking was performed on all input variables x_i in [13], [15]. However, this increases the complexity of the algorithm significantly and is not a practical realization. Therefore, in this study, the similarity check is performed only on the first input variable, that is, the speed error. The similarity measure E between the new membership function and all existing ones are calculated and the maximum one, $E_{\max}$, is found as follows:

$$E_{\max} = \max_{1 \le j \le M(t)} E\left(u\left(m_1^{(\text{new})}, \sigma_1^{(\text{new})}\right), u(m_{j1}, \sigma_{j1})\right) \quad (13)$$

where $u(m_{j1}, \sigma_{j1})$ represents the Gaussian membership function defined as (4) with mean m_{ji} and standard deviation σ_{ji}; $M(t)$ is the number of membership functions of the ith input variable. If $E_{\max} \le \bar{E}$, where $\bar{E} \in (0,1)$ is a prespecified value, then the new membership function is adopted and the number $M(t)$ is incremented

$$M(t+1) = M(t) + 1. \quad (14)$$

Since the generation of a membership function corresponds to the generation of a new fuzzy rule, the link weight, $\omega^{(\text{new})}$, associated with a new fuzzy rule has to be decided. Generally, the link weight $\omega^{(\text{new})}$ is selected with random or prespecified constant.

2) Parameter Learning Phase: The central part of the parameter learning algorithm for the SCFNN is to obtain adaptive rules to adjust the parameters of the network based on a given set of input-output pairs. If the parameters of the network are considered as elements of a parameter vector, the learning process involves the determination of the vector which minimize a given energy function. The gradient of the energy function with respect to the vector is computed and the vector is adjusted along the negative gradient. This method is generally referred to as the backpropagation learning rule because the gradient vector is calculated in the direction opposite to the flow of the output of each node. To describe the online parameter learning algorithm of the SCFNN using the supervised gradient descent method, first the energy function E is defined as

$$E = \frac{1}{2}(\omega_m - \omega_r)^2 = \frac{1}{2}e_m^2. \tag{15}$$

Then the parameter learning algorithm based on backpropagation is described in the following:

Layer 4: The error term to be propagated is computed as

$$\delta^4 = -\frac{\partial E}{\partial y^*} = \left[-\frac{\partial E}{\partial e_m}\frac{\partial e_m}{\partial y^*}\right] = \left[-\frac{\partial E}{\partial e_m}\frac{\partial e_m}{\partial \omega_r}\frac{\partial \omega_r}{\partial y^*}\right] \tag{16}$$

and the link weight is updated by the amount

$$\Delta w_j = -\eta_w \frac{\partial E}{\partial w_j} = \left[-\eta_w \frac{\partial E}{\partial y^*}\right]\left(\frac{\partial y^*}{\partial w_j}\right) = -\eta_w \delta^4 u_j \tag{17}$$

where factor η_w is the learning-rate parameter of the link weight. The link weights in layer 4 are updated according to the following equation:

$$w_j(N+1) = w_j(N) + \Delta w_j \tag{18}$$

where N denotes the iteration number of the jth link.

Layer 3: In this layer only the error term needs to be calculated and propagated

$$\delta_j^3 = -\frac{\partial E}{\partial u_j} = \left[-\frac{\partial E}{\partial y^*}\right]\left[\frac{\partial y^*}{\partial u_j}\right] = \delta^4 w_j \tag{19}$$

Layer 2: The error term is computed as follows:

$$\delta_{ji}^2 = -\frac{\partial E}{\partial u_{A_i^j}} = \left[\frac{\partial E}{\partial y^*}\frac{\partial y^*}{\partial u_j}\right]\left[\frac{\partial u_j}{\partial u_{A_i^j}}\right] = \delta_j^3 \tag{20}$$

The update law of m_{ji} is

$$\begin{aligned}\Delta m_{ji} &= -\eta_m \frac{\partial E}{\partial m_{ji}} = \left[-\eta_m \frac{\partial E}{\partial u_{A_i^j}}\frac{\partial u_{A_i^j}}{\partial m_{ji}}\right]\\ &= \eta_m \delta_{ji}^2 \frac{2\left(x_i^2 - m_{ji}\right)}{(\sigma_{ji})^2}\end{aligned} \tag{21}$$

The update law of σ_{ji} is

$$\begin{aligned}\Delta \sigma_{ji} &= -\eta_\sigma \frac{\partial E}{\partial \sigma_{ji}} = \left[-\eta_\sigma \frac{\partial E}{\partial u_{A_i^j}}\frac{\partial u_{A_i^j}}{\partial \sigma_{ji}}\right]\\ &= \eta_\sigma \delta_{ji}^2 \frac{2\left(x_i^2 - m_{ji}\right)^2}{(\sigma_{ji})^3}\end{aligned} \tag{22}$$

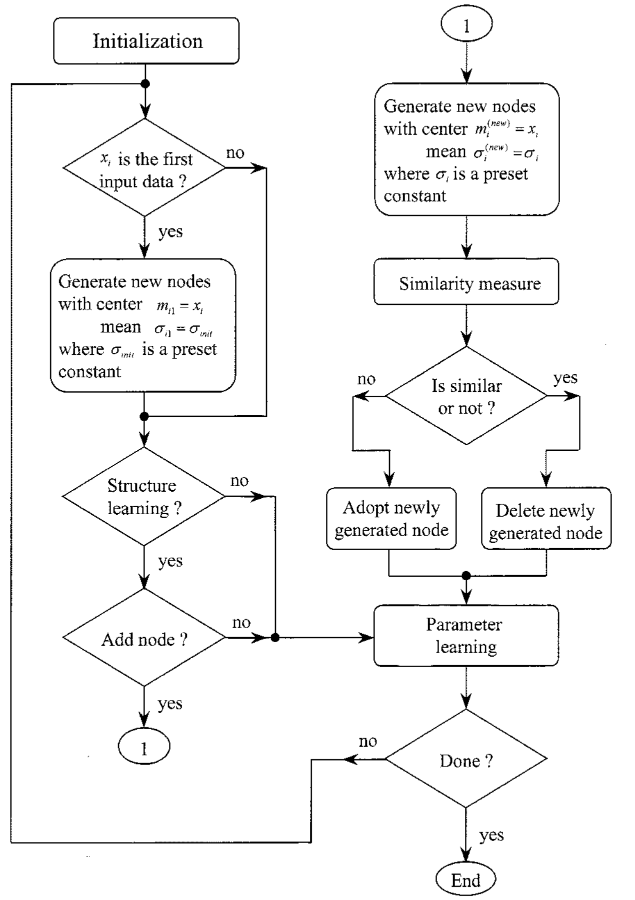

Fig 3. Flow diagram of structure and parameter learning algorithms for SCFNN.

where η_m and η_σ are the learning-rate parameters of the mean and the standard deviation of the Gaussian function, respectively. The mean and standard deviation of the membership functions in this layer are updated as following:

$$m_{ji}(N+1) = m_{ji}(N) + \Delta m_{ji} \tag{23}$$

$$\delta_{ji}(N+1) = \delta_{ji}(N) + \Delta \delta_{ji}. \tag{24}$$

The exact calculation of the Jacobian of the system $\partial\omega_r/\partial y^*$ which is contained in $\partial E/\partial y^*$ cannot be determined due to the uncertainties of the plant dynamic such as parameter variations and external disturbances. To overcome this problem and to increase the online learning rate of the network parameters, the delta adaptation law proposed in [16] as follows is adopted:

$$\delta^4 \equiv A e_m + \dot{e}_m \tag{25}$$

where A is a positive constant. The detailed proof of the convergence of the SCFNN using (25) is similar to [16] and is omitted here. The flow diagram of the learning algorithms of the SCFNN is shown in Fig. 3.

IV. Simulation and Experimental Results

A. Simulation of SCFNN Speed Controller

To investigate the effectiveness of the proposed SCFNN speed controller, a second-order system transfer function with the following prescribed characteristics: 0.3 s rise time, no steady-state error and unity damping ratio to avoid overshoot is

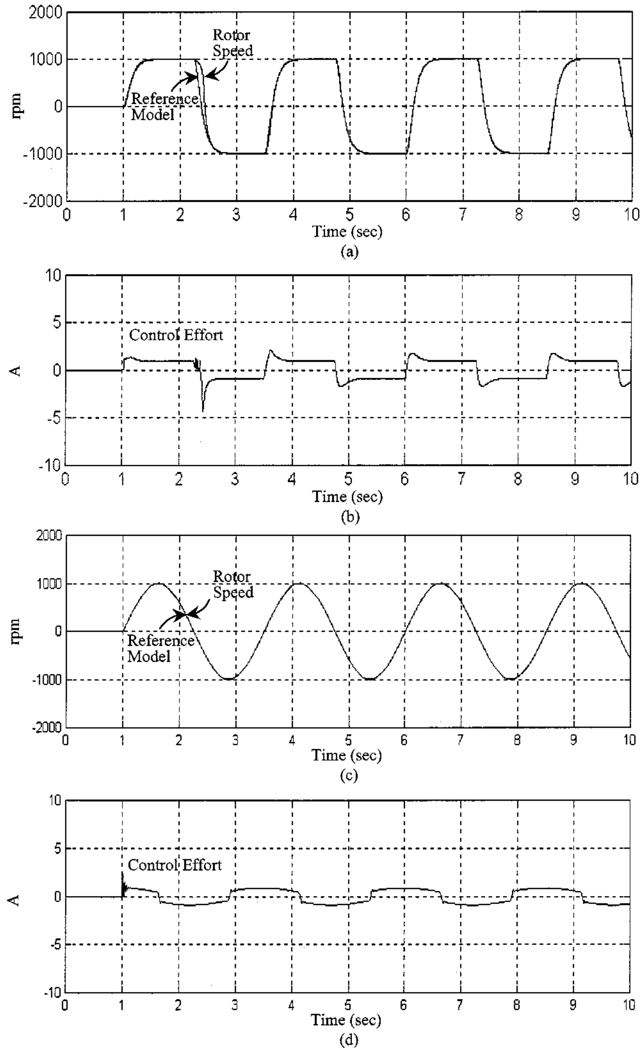

Fig. 4. Simulation results of SCFNN speed control system at nominal condition. (a) Speed response for periodic step command. (b) Control effort for periodic step command. (c) Speed response for periodic sinusoidal command. (d) Control effort for periodic sinusoidal command.

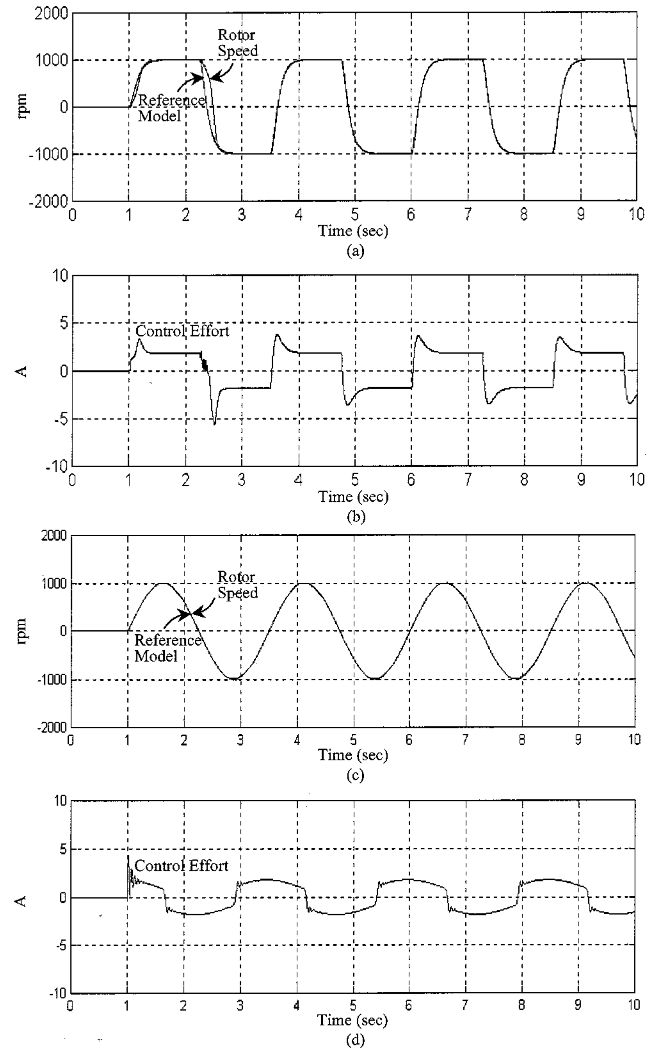

Fig. 5. Simulation results of SCFNN speed control system at rotor inertia variation condition. (a) Speed response for periodic step command. (b) Control effort for periodic step command. (c) Speed response or periodic sinusoidal command. (d) Control effort for periodic sinusoidal command.

chosen as the reference mode l for the periodic step command. Therefore, in accordance with [17], the following transfer function is chosen in this study:

$$G(s) = \frac{168.11}{s^2 + 25.93s + 168.11} \tag{26}$$

Moreover, when the command input is a sinusoidal reference trajectory, the reference model shown in Fig. 1 is set to be one. Two conditions of rotor inertia are tested here, one being the nominal condition, and the other being the increasing of the rotor inertia to 3 times the nominal value. The simulation is implemented based on the scheme shown in the Fig. 1 and the speed control loop and the training of the SCFNN are carried out every 2 ms. The simulation is carried out using the Matlab package. The initial formation procedure [11], which is aimed at extracting the structure, the connective weights and the membership functions from the process of the interative learning, is used to initial the parameters of the SCFNN to speed up the convergence process. To test the command tracking performance of the proposed motor drive system, the simulation results of the rotor speed and the associated control effort at nominal condition for periodic step and sinusoidal commands changing from 1000 rpm to -1000 rpm are shown in Figs. 4(a), (b), and (c), (d), respectively. The simulation results at the condition of rotor inertia variation are shown in Fig. 5 with the same simulation conditions as nominal condition. The responses of the rotor speed and the associated control effort for periodic step and sinusoidal commands changing from 1000 rpm to -1000 rpm are shown in Figs. 5(a), (b), and (c), (d), respectively. From the simulation results of command tracking, after one command cycle of online structure and parameters learning, precise tracking response is obtained for both the nominal and parameter variation conditions. Therefore, the robust tracking performance of the SCFNN is obvious. Furthermore, to test the load regulated control performance of the motor drive system, the simulation result of a step-load disturbance with 1 nm at nominal condition and the rotor inertia variation condition for constant speed 1000 rpm are shown in Fig. 6(a) and (b). From the simulation results, excellent load regulation ability of the SCFNN at the condition of parameter variation and external load disturbance can be observed clearly.

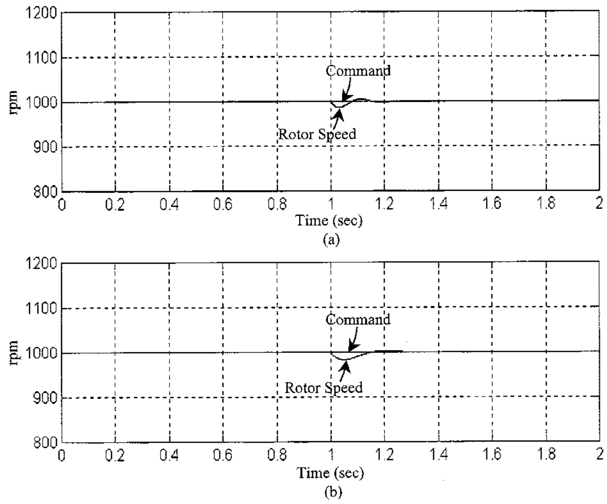

Fig. 6. Simulation results of SCFNN speed control system for disturbed torque regulation. (a) Nominal condition. (b) Rotor inertia variation condition.

B. *Experimentation of SCFNN Speed Controller*

Using the same conditions as in the simulation, experimental results using a physical plant are provided here to demonstrate the effectiveness of the proposed SCFNN speed controller. The proposed control system is implemented using a Pentium computer with Turbo C package. The sampling rate of the SCFNN controller including the online training algorithm is 2 ms. The initial formation procedure is also used to initial the parameters of the SCFNN. The measured responses of the rotor speed and the associated control effort for the periodic step and sinusoidal commands at nominal condition are shown in the Fig. 7. Since both the structure and parameters of the SCFNN are learned online, accurate tracking performance can be obtained after one cycle of step and sinusoidal commands for periodic step and sinusoidal command inputs. Moreover, the measured responses of the rotor speed and the associated control effort at the condition of increasing the rotor inertia to approximately 3 times the nominal value for the periodic step and sinusoidal commands are shown in Fig. 8. Then, the measured responses of a step-load disturbance with 1 nm at nominal condition and the rotor inertia variation condition for constant speed 1000 rpm are shown in Fig. 9. From the experimental results shown in Figs. 8 and 9, the robustness of the motor drive system using SCFNN controller at the conditions of constant parameter variation and external disturbance is obvious after the network being well trained.

C. *Simulation of SCFNN Speed Controller Considering Time-Varying and Nonlinear Disturbances*

To further investigate the ability and dynamic characteristics of the proposed SCFNN controller over a broad operating range, three simulation cases including time-varying parameter variation, time-varying external load disturbance, and nonlinear friction force in the shaft are considered as follows:

$$\text{Case 1: } a = 4.4\sin(\omega t) \quad T_L = 0 \tag{27}$$

$$\text{Case 2: } J = \bar{J} \quad B = \bar{B}$$
$$T_L = \sin(\omega t)\text{nm} \quad \text{at } t \geq 3.5 \text{ s} \tag{28}$$

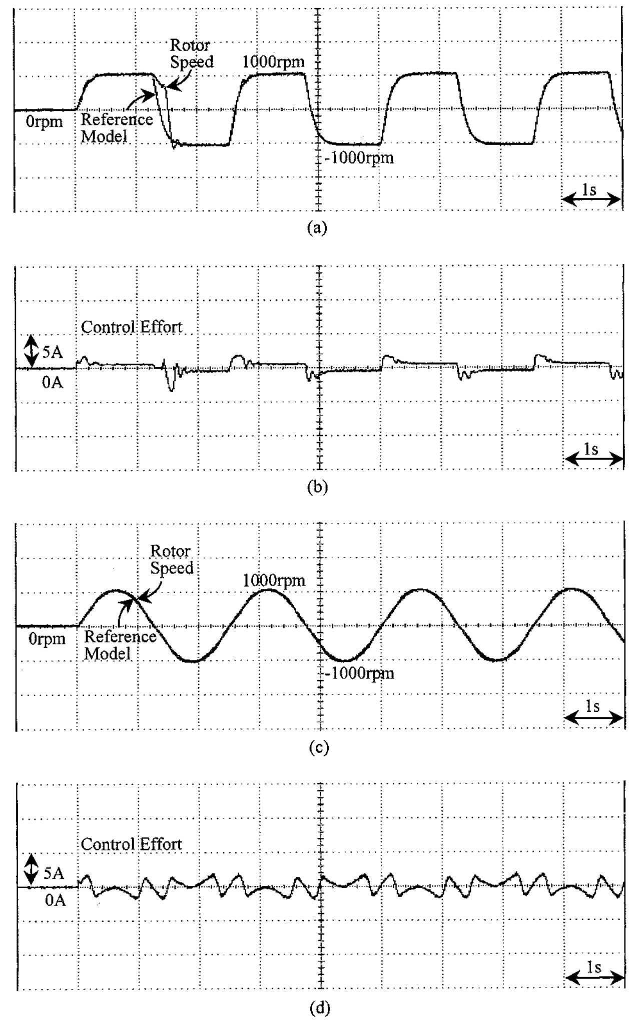

Fig. 7. Measured responses of SCFNN speed control system at nominal condition. (a) Speed response for periodic step command. (b) Control effort for periodic step command. (c) Speed response or periodic sinusoidal command. (d) Control effort for periodic sinusoidal command.

$$\text{Case 3: friction force } f(\omega_r) = F_C\,\text{sgn}(\omega_r) + (F_S + F_C)e^{-(\omega_r/v_S)^2}\,\text{sgn}(\omega_r) + K_v\omega_r \tag{29}$$

In Case 3, F_C is the Coulomb friction; F_S is the static friction; v_S is the Stribeck velocity parameter [18]; K_v is the coefficient of viscous friction; $\text{sgn}(\cdot)$ is a sign function. The coefficients of the friction model used in this study are selected as follows:

$$F_C = 1.1 \quad F_S = 1.2 \quad v_S = 1 \quad K_v = 10. \tag{30}$$

To investigate the dynamic of the SCFNN control system, the initial formation procedure is not adopted in these three simulation cases. Moreover, the initial rule number is zero. The simulation is also implemented based on the scheme shown in the Fig. 1 with the speed control loop and the training of the SCFNN being carried out every 2 ms. The simulation results of the rotor speed and the associated control effort at Case 1 for periodic step and sinusoidal commands changing from 1000 rpm to -1000 rpm are shown in Fig. 10(a), (b), and (c), (d), respectively. The

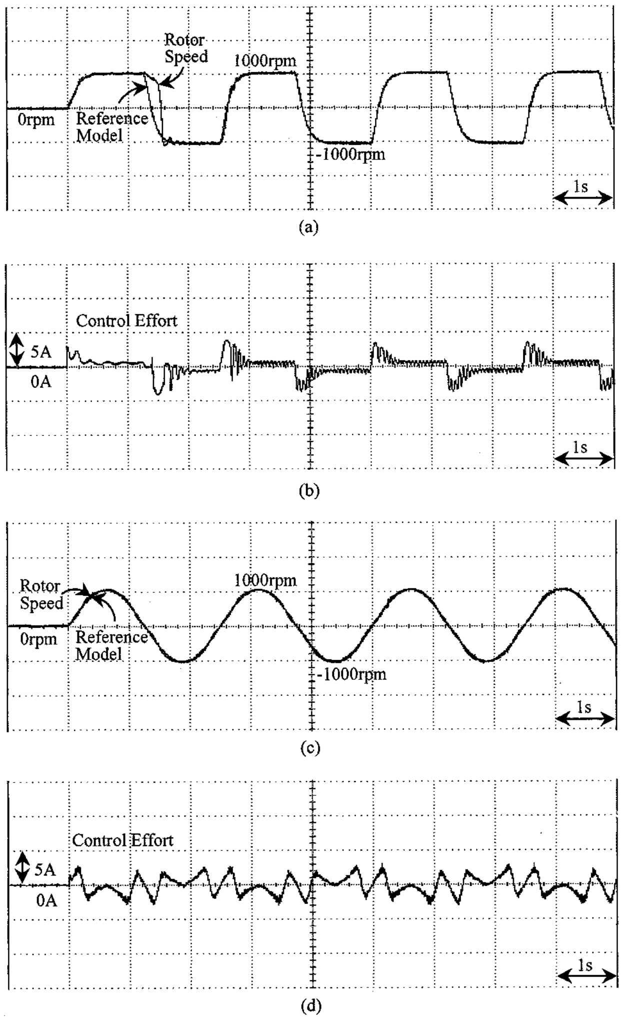

Fig. 8. Measured responses of SCFNN speed control system at rotor inertia variation condition. (a) Speed response for periodic step command. (b) Control effort for periodic step command. (c) Speed response or periodic sinusoidal command. (d) Control effort for periodic sinusoidal command.

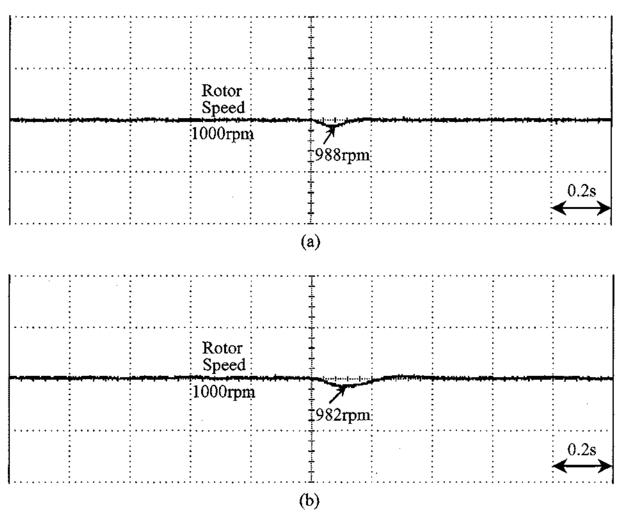

Fig. 9. Measured responses of SCFNN speed controller for disturbed torque regulation. (a) Nominal condition. (b) Rotor inertia variation condition.

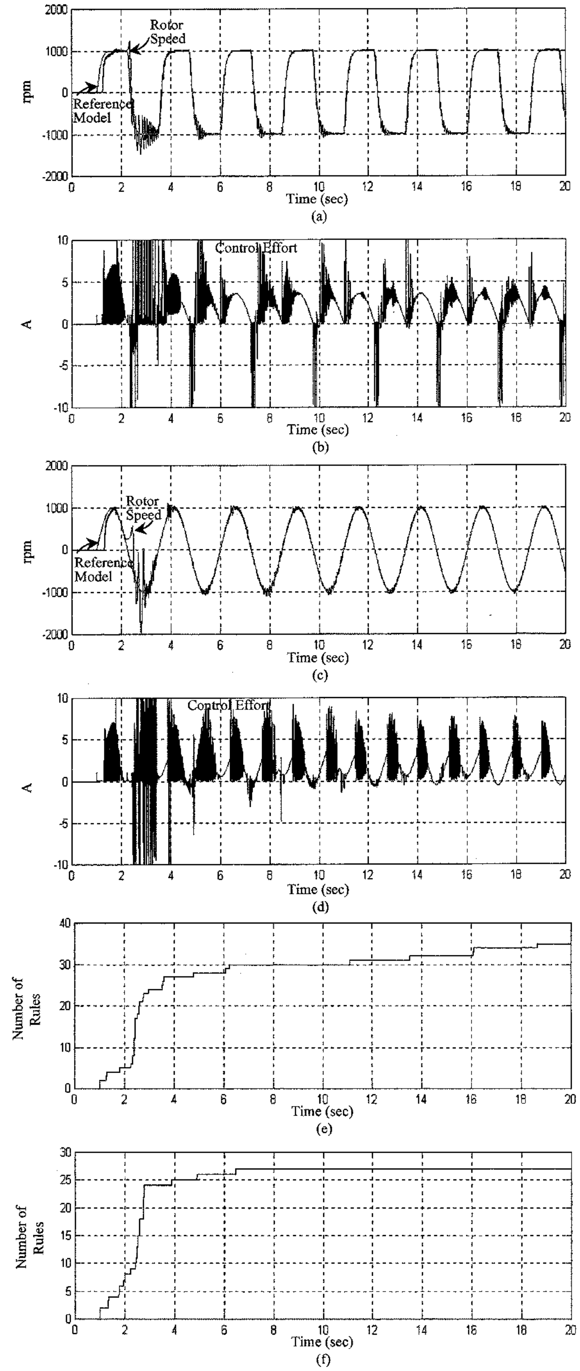

Fig. 10. Simulation results of SCFNN speed controller at Case 3. (a) Speed response for periodic step command. (b) Control effort for periodic step command. (c) Speed response for periodic sinusoidal command. (d) Control effort for periodic sinusoidal command. (e) Changing of number of rules for periodic step command. (f) Changing of number of rules for periodic sinusoidal command.

number of rules changing throughout the learning phase at Case 1 for periodic step and sinusoidal commands are shown in Fig. 10(e) and (f). Since the dynamic of periodic step command input is more complicated than the dynamic of periodic sinusoidal input, for periodic step command input the number of rules increases to 35 in 20 s, and for periodic sinusoidal command input the number of rules increases to 27 in 20 s. Because there is only input and output nodes of the SCFNN initially, favorable tracking performance for the periodic step and sinusoidal commands can be obtained after a few cycles online training of the SCFNN at the time-varying parameter condition.

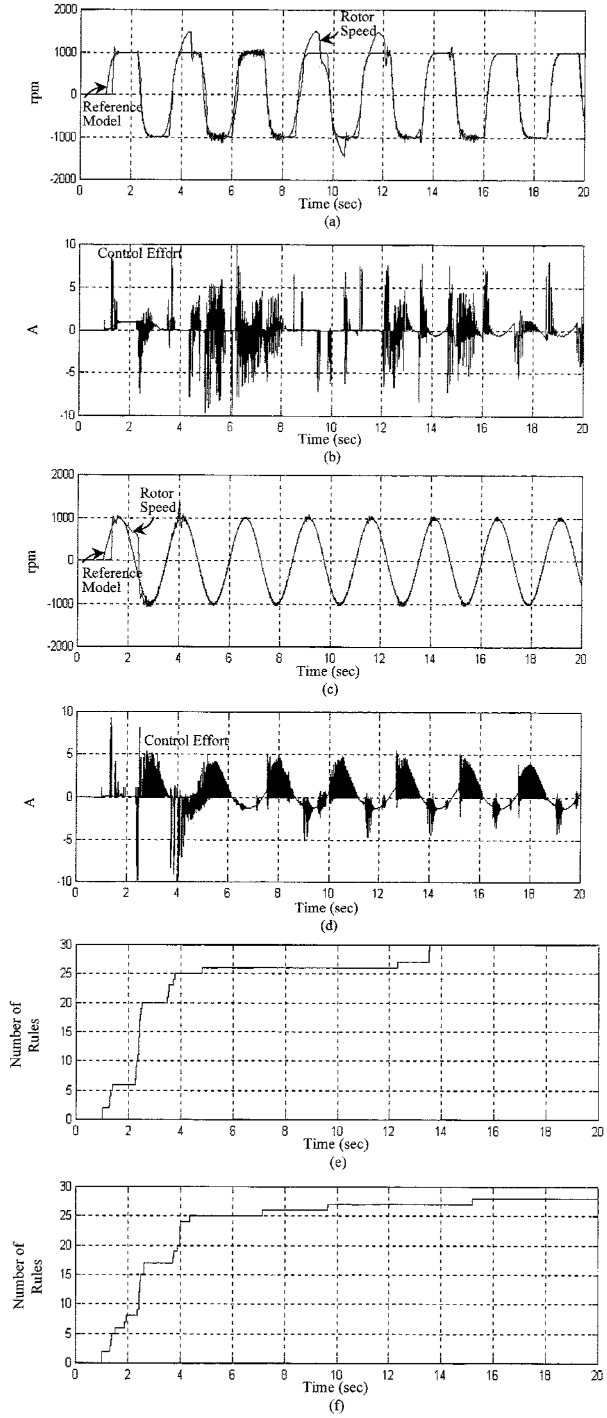

Fig. 11. Simulation results of SCFNN speed controller at Case 2. (a) Speed response for periodic step command. (b) Control effort for periodic step command. (c) Speed response for periodic sinusoidal command. (d) Control effort for periodic sinusoidal command. (e) Changing of number of rules for periodic step command. (f) Changing of number of rules for periodic sinusoidal command.

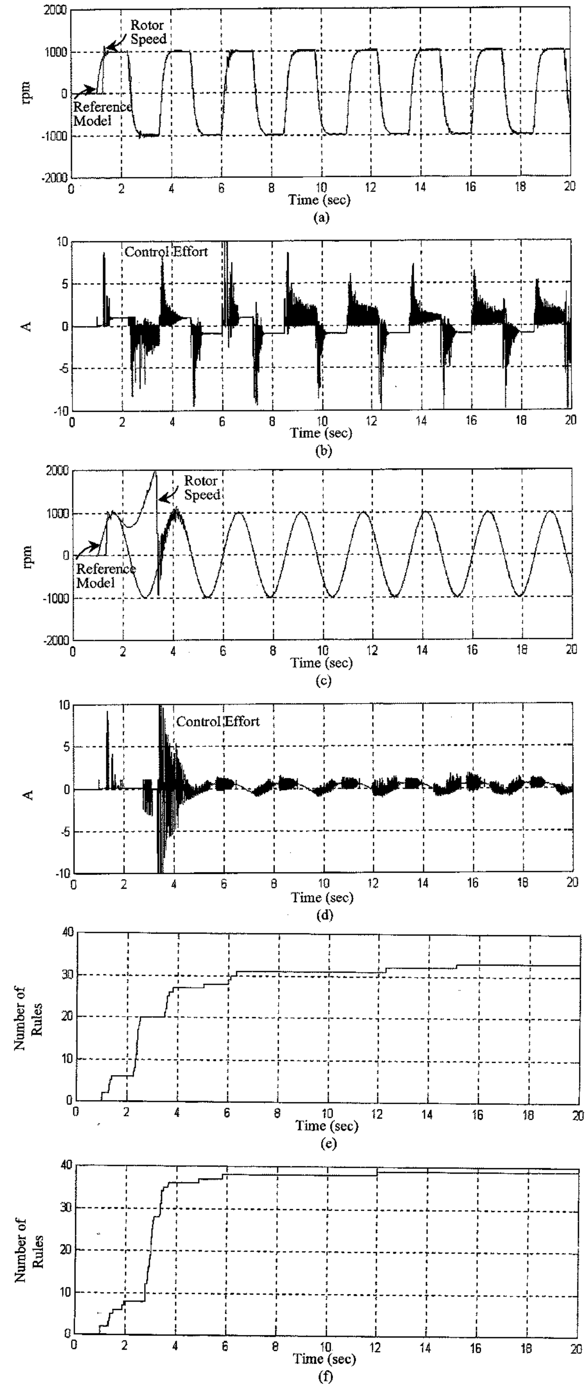

Fig. 12. Simulation results of SCFNN speed controller at Case 3. (a) Speed response for periodic step command. (b) Control effort for periodic step command. (c) Speed response for periodic sinusoidal command. (d) Control effort for periodic sinusoidal command. (e) Changing of number of rules for periodic step command. (f) Changing of number of rules for periodic sinusoidal command.

Furthermore, the simulation results of the rotor speed and the associated control effort at Case 2 for the periodic step and sinusoidal commands changing from 1000 rpm to -1000 rpm are shown in Fig. 11, (b), and (c), (d), respectively. The number of rules changing throughout the learning phase at and Case 2 for periodic step and sinusoidal commands are shown in Fig. 11(e) (f). Initially, there is only input and output nodes of the SCFNN, thus good tracking performance can be obtained after a few cycles online training of the SCFNN at the time-varying external disturbance condition. Robust control performance of the SCFNN speed controller in the condition of time-varying external load disturbance is obvious. In addition, the simulation results of the rotor speed and the associated control effort at Case 3 for the periodic step and sinusoidal commands changing from 1000 rpm to -1000 rpm are shown in Fig. 12(a), (b), and (c), (d), respectively. The number of rules changing throughout the learning phase at Case 3 for periodic step and sinusoidal commands are shown in Fig. 12(e) and (f). The dynamic influence of the sign functions in the friction force on the periodic sinusoidal command input is more than there influence on the periodic sinusoidal input. Thus, for periodic step command input, the number of rules increases to 33 in 20 s and for periodic sinusoidal command input the number of rules increases to 39 in 20 s. Since there is only input and output nodes of the SCFNN initially, accurate and robust tracking performance for the periodic step and sinusoidal command input can be obtained after three cycles online training of the SCFNN at the nonlinear friction force condition.

V. Conclusion

A self-constructing type fuzzy neural network is successfully developed for practical implementation in this study. Initially, there are only input and output nodes in the proposed SCFNN, then the structure and parameter learning are done concurrently and online. First, the structure of the network and the training algorithms were described in detail. Next, a PMSM speed drive system was adopted to verify the effectiveness of the proposed control stratagem. Finally, the control performance of the proposed SCFNN controller over a broad operating range has been demonstrated in both the simulation and experimentation.

References

[1] C. C. Lee, “Fuzzy logic in control systems: Fuzzy logic controller—Part I and part II,” *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 404–436, 1990.
[2] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*, New York: Wiley, 1994.
[3] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[4] T. Fukuda and T. Shibata, “Theory and applications of neural networks for industrial control systems,” *IEEE Trans. Ind. Electron.*, vol. 39, pp. 472–491, 1992.
[5] P. S. Sastry, G. Santharam, and K. P. Unnikrishnan, “Memory neuron networks for identification and control of dynamical systems,” *IEEE Trans. Neural Networks*, vol. 5, pp. 306–319, 1994.
[6] C. C. Ku and K. Y. Lee, “Diagonal recurrent networks for dynamic systems control,” *IEEE Trans. Neural Networks*, vol. 6, pp. 144–156, 1995.
[7] S. Horikawa, T. Furuhashi, and Y. Uchikawa, “On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm,” *IEEE Trans. Neural Networks*, vol. 3, pp. 801–806, 1992.
[8] Y. C. Chen and C. C. Teng, “A model reference control structure using a fuzzy neural network,” *Fuzzy Sets and Syst.*, vol. 73, pp. 291–312, 1995.
[9] T. S. R. Jang and C. T. Sun, “Neural-fuzzy modeling and control,” *Proc. IEEE*, vol. 83, pp. 378–405, 1995.
[10] F. J. Lin, R. F. Fung, and R. J. Wai, “Comparison of sliding mode and fuzzy neural network control for motor-toggle servomechanism,” *IEEE/ASME Trans. Mechatron.*, vol. 3, pp. 302–318, 1998.
[11] F. J. Lin, W. J. Huang, and R. J. Wai, “A supervisory FNN control system for tracking periodicic inputs,” *IEEE Tran. Fuzzy Syst.*, vol. 7, pp. 41–52, Feb. 1999.
[12] C. T. Sun, “Rule-based structure identification in an adaptive-network-based fuzzy inference,” *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 46–63, 1994.
[13] C. T. Lin, “A neural fuzzy control system with structure and parameter learning,” *Fuzzy Sets Syst.*, vol. 70, pp. 183–212, 1995.
[14] C. T. Lin and C. S. G. Lee, *Neural Fuzzy System: A Neural-Fuzzy Synergism to Intelligent System*. Englewood cliffs, NJ: Prentice-Hall, 1996.
[15] C. F. Juang and C. T. Lin, “An on-line self-constructing neural fuzzy inference network and its application,” *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 12–32, Feb. 1998.
[16] F. J. Lin, R. J. Wai, and C. C. Lee, “Fuzzy neural network position controller for ultrasonic motor drive using push-pull DC-DC converter,” in *Inst. Elect. Eng. Proc. Contr. Theory Appl.*, vol. 146, 1999, pp. 99–107.
[17] F. J. Lin, “Robust speed-controlled induction-motor drive using EKF and RLS estimators,” *Inst. Elect. Eng. Proc. Electron. Power Appl.*, vol. 143, pp. 186–192, 1996.
[18] C. C. de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, “A new model for control of systems with friction,” *IEEE Tran. Automat. Contr.*, vol. 40, pp. 419–425, 1995.