Computer Science and Information Systems 21(2):569–592 https://doi.org/10.2298/CSIS230803076C

Robust Compensation with Adaptive Fuzzy Hermite Neural Networks in Synchronous Reluctance Motors

Chao-Ting Chu[1] and Hao-Shang Ma[2,⋆]

[1] Chunghwa Telecom Laboratories, Internet of Things Laboratory,
No.99, Dianyan Rd., Yangmei District, Taoyuan City 32661, Taiwan, R. O. C.
chaot@cht.com.tw
[2] Department of Computer Science and Information Engineering,
National Taichung University of Science and Technology,
No. 129, Section 3, Sanmin Road, North District, Taichung City 404336, Taiwan, R. O. C.
hsma@nutc.edu.tw

Abstract. In this paper, a robust compensation scheme using adaptive fuzzy Hermite neural networks (RCAFHNN), for use in synchronous reluctance motors (SRMs), is proposed. SRMs have a simple underlying mathematical model and mechanical structure, but are affected by problems related to parameter variations, external interference, and nonlinear dynamics. In many fields, precise control of motors is required. Although the use of neural network and fuzzy are widespread, such controllers are affected by unbound nonlinear system model. In this study, RCAFHNN, based on an adaptive neural fuzzy interface system (ANFIS), was used to bound motor system model controller algorithm. RCAFHNN can be characterized in three parts. First, RCAFHNN offers fuzzy expert knowledge, a neural network for online estimation, and recursive weight estimation. Second, the replacement of the Gaussian function by the Hermite polynomial in RCAFHNN enables reduced membership function training times. Third, the system convergence and robustness compensation of RCAFHNN were confirmed using Lyapunov stability. RCAFHNN ameliorates the problems of external load and system lump uncertainty. The experimental results, in which the output responses of RCAFHNN and ANFIS (adaptive neural fuzzy interface systems) were compared, demonstrated that RCAFHNN exhibited superior performance.

Keywords: Synchronous reluctance motors, Lyapunov stability, Robust, Adaptive control, Neural network estimator, Adaptive laws.

1. Introduction

In recent years, motor control has gained significant popularity [6, 16, 19, 23]. A three-phase motor is typically supplied by a three-phase AC power source. This means there are three cables providing power, and each cable's voltage has a phase difference of 120 degrees. These three phases are referred to as Phase A, Phase B, and Phase C. However, calculating the three-phase system involves complex mathematical equations and issues related to mutual inductance coupling. Traditionally, we can employ coordinate transformation to convert the system from three-phase to two-phase, simplifying the calculations and also addressing the mutual inductance coupling issues associated with the

⋆ Corresponding author

motor. It need to address various uncertainties generated during the actual operation of the motor. Therefore, controlling alternating current in synchronous reluctance motors (SRMs) [1, 2, 8] has become a central concern. SRMs have a simple underlying mathematical model and mechanical structure but are affected by nonlinear problems such as parameter variations, external loads, and nonlinear friction. Numerous studies have explored using the controller to mitigate these problems such as robust control [27]. Moreover, a instead controller is used for controlling a SRMs using Hermite neural networks. Hermite polynomials replace traditional Gaussian functions, eliminating the need to select the vertices and widths of Gaussian functions, thereby simplifying the computational complexity. Additionally, recursive weights are employed to increase the parameters of the neural network. The Lyapunov method is used to prove that the system overcomes the cumulative uncertainties, ensuring the stability of the motor system control.

Inspiring the success of deep learning on many fields, various neural network structures have been proposed [5, 7, 10, 13, 15, 18, 28]. These research utilize the non-linear capabilities of neural network to learn and adapt to auto control. For example, an adaptive NN dynamic surface controller design for nonlinear pure-feedback switched systems with time-delays and quantized input, showing that the system's output response had satisfactory performance. A wavelet neural network sliding-mode controller [7] was used in a permanent magnet synchronous motor, where the width of the wavelet function improved neural network function. In addition, fuzzy controllers and neural networks each have distinctive advantages. Some studies have combined these two controllers to create adaptive neural fuzzy interface systems (ANFISs) [14,25,26]. ANFIS combines fuzzy expert knowledge with online neural network learning, resulting in better performance than using a simple fuzzy controller or neural network controller. In neural networks-based control systems, Gaussian functions are commonly employed. However, Gaussian functions have a limitation in that they rely on peak and width parameters, which necessitates more intricate calculations to ascertain the most suitable values for these parameters.

In this work, an RCAFHNN is proposed for use in SRMs, exhibiting satisfactory output responses in experimental results that include Laypunov functions to ensure system stability. Control inputs do not require nonlinear system parameters, and Hermite polynomials replace traditional Gaussian functions, eliminating the need to choose optimal vertices and widths. From the experimental results, we can observe that RCAFHNN offered satisfactory performance in handling lumped uncertainty and nonlinear dynamics.

The main contributions of this work are as follows.

- We propose a controller which utilizes Hermite neural networks to control synchronous reluctance motors. Hermite polynomials replace traditional Gaussian functions to simplify the computational complexity.
- A recursive weighting is used to increase neural network parameters in AFHNN, and a Lyapunov-based approach is employed to demonstrate the system's ability to overcome total uncertainty, ensuring stable control of the motor system.
- Experimental tests are conducted under various challenging conditions, including unloaded, loaded, and rotational wave commands, to evaluate the performance of the proposed controller.

The remainder of this paper is organized as follows. The mathematical model of the SRM system is presented in Section 3. The RCAFHNN is described in Section 4. The experimental results are shown in Section 5, and they demonstrate that the proposed

RCAFHNN offers fast performance and satisfactory dynamic responses when handling parameter variations and external loads. Finally, the conclusion is presented in Section 6.

2. Related Works

Studies have sought to improve the stability of nonlinear systems in robust control with neural networks [9–12]. Hsiao et al. [10] employed a neural-network-based approach with delay-dependent robust stability criteria, and they analyzed dithered chaotic systems with multiple time-delays. Huang et al. [11] presented an evolutionary radial basis function neural network combined with robust genetic-based immune computing, achieving precise command tracking in autonomous robots. In the field of motors, precise position control of sensorless PMSM [12] servo drives is required. Adaptive robust speed control with a recurrent Elman neural network can offer more precise control of a system and decrease system position errors. Gong et al. [9] also proposed robust state estimation for delayed complex-valued neural networks to consider available output measurements containing nonlinear Lipschitz-like terms.

Work environments demand precise control of drilling machines [4,29]. Self-optimizing algorithms [4] and switched-control algorithms [29] have been employed in pressure drilling and have demonstrated satisfactory performance results. Viola et al. [22] also propose a parallel enabled and stability-aware self optimizing control for using numerical twin instances during the most computationally intensive steps. Several studies have investigated fuzzy neural network sliding-mode controllers [3, 10, 30]. Fuzzy neural networks can reduce the system chattering phenomenon and can train parameters online to increase the precision of the system. Castaneda et al. [3] and Song et al. [21] used neural sliding-mode controllers in motors, and online neural network training enabled the system to overcome lumped uncertainties.

Various neural network structures have been proposed [5, 7, 10, 13, 15, 18, 28]. Hsiao et al. [10] proposed a neural-network-based approach for delay-dependent robust stability criteria for dithered chaotic systems with multiple time-delays. Niu et al. [18] proposed an adaptive NN dynamic surface controller design for nonlinear pure-feedback switched systems with time-delays and quantized input, showing that the system's output response had satisfactory performance. Additionally, Chen et al. [5] researched a rotor fault diagnosis system based on sGA-based individual neural networks, utilizing GA algorithms to search for optimal parameters to address nonlinear system issues.

A wavelet neural network sliding-mode controller [7] was used in a permanent magnet synchronous motor, where the width of the wavelet function improved neural network function. Yin et al. [24] used a Hermite neural network as an activation function. Similarly to the wavelet function, the width of the Hermite function enabled satisfactory system performance. Studies have also utilized diagonal neural networks with second-order learning algorithms [13] in system identification [20, 28] due to the faster convergence speed of second-order algorithms compared to that of first-order algorithms.

Fuzzy controllers and neural networks each have distinctive advantages. Some studies have combined these two controllers to create adaptive neural fuzzy interface systems (ANFISs) [14, 25, 26]. Yun et al. used RBFNN and ANFIS to predict the market price of electricity [25] and demonstrated that ANFIS offered accurate predictions. The power amplifier modeling conducted in [26] incorporated ANFIS to identify various effects and

different rules. Liu et al. proposed a new ANFIS structure [14] using numerical analysis and classification.

3. SynRM mathematical model

The voltage equations of the $d-q$ axis equivalent architecture in a SynRM are expressed as

$$V_{ds} = R_s i_{ds} - \omega_R L_{qs} i_{qs} + L_{ds} \frac{di_{ds}}{dt} \tag{1}$$

$$V_{qs} = R_s i_{qs} - \omega_R L_{ds} i_{ds} + L_{qs} \frac{di_{qs}}{dt} \tag{2}$$

where V_{ds} and V_{qs} are the direct and quadrature axis voltages, respectively. i_{ds} and i_{qs} are the direct and quadrature axis currents, respectively. L_{ds} and L_{qs} are the direct and quadrature inductances, respectively. R_s is the copper loss resistor. ω_R is the rotor velocity in SynRM.

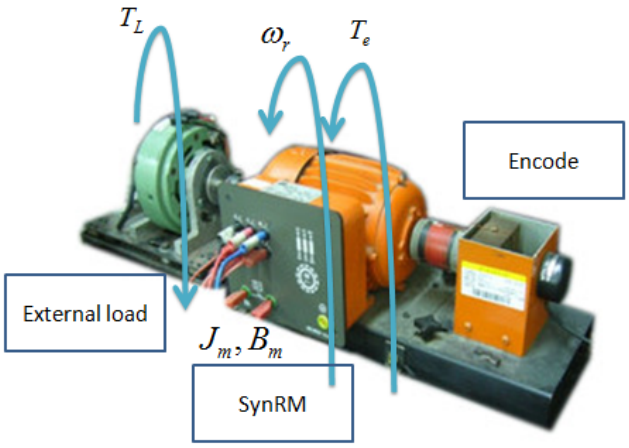

Fig. 1. Torque architecture of SRM

The torque architecture of SynRM in the mechanical equation that shows in Figure 1, and the equation is expressed as

$$T_e = J_m \frac{d\omega_R}{dt} + B_m \omega_R + T_L \tag{3}$$

where T_e is the torque of SynRM, T_L is the external load of torque, J_m is the moment of inertia, B_m is the coefficient of friction. We can rewrite the dynamic equation 3 as

$$\dot{\omega}_r = -\frac{B_m}{J_m} \omega_R + \frac{1}{J_m}(T_e - T_L) \tag{4}$$

We have electromagnetic torque equation in the rotating $d-q$ reference axis as

$$T_e = \frac{3}{4}P(L_{ds} - L_{qs})i_{ds}i_{qs} \tag{5}$$

where P is the poles in the SynRM. Therefore, the system model of SynRM is showed in Figure 2.

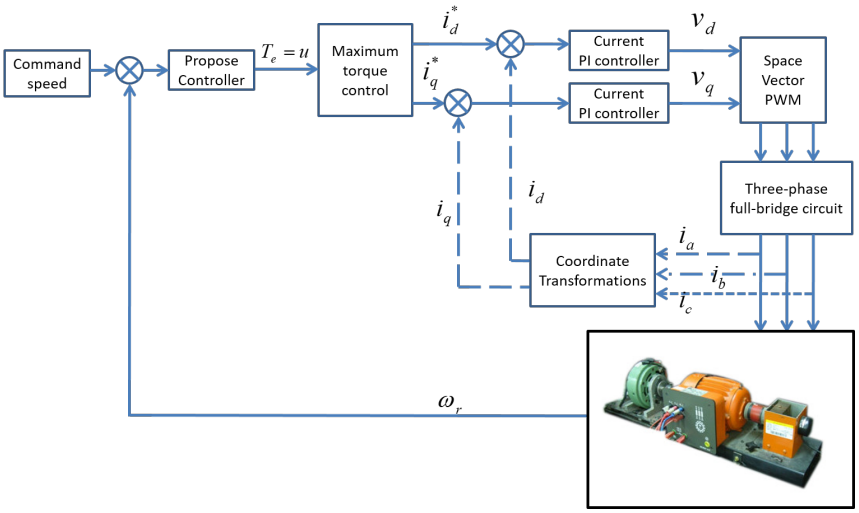

Fig. 2. Gaussian basis function neural network

4. Design of robust compensation with adaptive fuzzy Hermite neural network

4.1. SynRM nonlinear system equation

Consider a nonlinear system equation as

$$\dot{x} = f(x) + b(x)u, \tag{6}$$

$$y = x, \tag{7}$$

where $f(x)$ and $b(x)$ are unknown real continuous nonlinear functions, $u \in R$ is the control input, $y \in R$ is the system output, and $x = \omega_d \in R$ is the state vector of the system, which we assume to be available for measurement. In order to be controllable for the dynamic system, function $b(x)$ must be nonzero for vector x in certain controllability regions. Without loss of generality, we assume that $0 < b(x) < \infty$. We can rewrite the dynamic equation 6 as

$$\dot{x} = f_1(x) + b_1(x)u + E(x) \tag{8}$$

where $f(x) = f_1(x) + f_2(x)$, $b(x) = b_1(x) + b_2(x)$, $E(x) = f_2(x) + f_2(x)$, $f_1(x)$ and $b_1(x)$ are the known real continuous parameters. $f_2(x)$ and $b_2(x)$ are the unknown real continuous parameters.

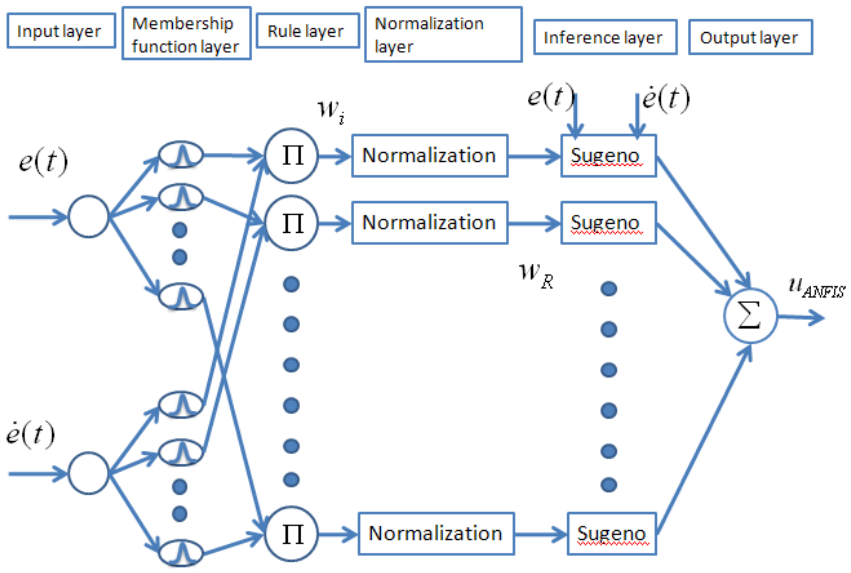

Fig. 3. The structure of ANFIS

4.2. Adaptive neural fuzzy inference system

ANFIS (Adaptive Neuro-Fuzzy Inference System) combines fuzzy expert knowledge with online neural network learning, resulting in better performance than using a simple fuzzy controller or neural network controller. The ANFIS structure is depicted in Figure 3, consisting of six layers. The first layer serves as the input layer, receiving the error signal into the network. This can be expressed by the equation:

$$y_1^1 = e(t), \quad y_2^1 = \dot{e}(t) \tag{9}$$

where $e(t) = x_d - x$, x_d is the command speed. Superscript is the n-th network, and subscript is n-th input.

The second layer is membership function layer, which is used fuzzificationcan to first layer, and the equation can be expressed as

$$y_j^2 = \exp\left[\frac{-(e(t) - v_j)^2}{2d_j^2}\right] \tag{10}$$

$$y_{j+\max j}^2 = \exp\left[\frac{-(\dot{e}(t) - v_{j+\max j})^2}{2d_{j+\max j}^2}\right] \tag{11}$$

where $\exp$ is the function of exponent, $\max j$ is the maxima of j, v_j is the Gauss function vertex, d_j is the Gauss function width, j is the j-th node.

The third layer is the rule layer, which is used logical product operator to second layer, so the output can be expressed as

$$y_i^3 = w_i = y_j^2 y_{j+\max j}^2, i = 1, 2, ..., Q \tag{12}$$

where Q is the rule number.

The fourth layer is normalization layer which is normalize to weight, and we can be expressed as

$$y_{R}^{4} = w_{R} = \frac{w_i}{\sum_{i=1}^{Q} w_i}, R = 1, 2, ..., Q \tag{13}$$

The fifth layer is the inference system, which is used Sugeno and average weighting method to defuzzification. The output can be expressed as

$$y_{R}^{5} = w_{R} f_{ANFIS}(e(t), \dot{e}(t)) = w_{R}(a_{R} e(t) + b_{R} \dot{e}(t) + c_{R}) \tag{14}$$

where $a_{R}, b_{R}, c_{R} > 0, R = 1, 2, ..., Q$ is the inference function.

The sixth layer is the output layer, which is used the linear combination of fifth layer, and the output can be expressed as

$$u_{ANFIS} = y^{6} = \sum_{R=1}^{Q} y_{R}^{5} \tag{15}$$

This paper is used the Lyapunov stability and steepest gradient method to convergence the network in ANFIS, in which we search optimal value of a_{R}, b_{R}, c_{R}. First define the Lyapunov function as

$$V_1 = \frac{1}{2} S^2 \tag{16}$$

where $S = h_1 \dot{e} + e, h_1 > 0$.

Stability criteria by the Lyapunov function, we must be $V < 0$, so that we has update equation of weight as follows

$$\Delta a_{R} = -\eta_{11} \frac{\partial \dot{V}_1}{\partial a_{R}} = -\eta_{11} \frac{\partial S \dot{S}}{\partial a_{R}} = -\eta_{11} \frac{\partial \dot{S}}{\partial a_{R}} \tag{17}$$

where η_{11} is the learning rate, $\eta_{11} > 0$,and we can rewrite equation 17 by calculus chain law as

$$\frac{\partial \dot{S}}{\partial a_{R}} = \frac{\partial \dot{S}}{\partial u_{ANFIS}} \frac{\partial u_{ANFIS}}{\partial a_{R}} \tag{18}$$

And equation 8 into equation 18, we obtain

$$\frac{\partial \dot{S}}{\partial u_{ANFIS}} = \frac{\partial h_1 \ddot{e} + \dot{e}}{\partial u_{ANFIS}} = \frac{\partial(\ddot{\omega}_d - f_1(x) - b_1(x) u_{ANFIS} - E(x) + h_0 \ddot{e})}{\partial u_{ANFIS}} = -b_1, \tag{19}$$

where $-b_{R} u_{ANFIS} = -b_{R} u$, and $\frac{\partial u_{ANFIS}}{\partial a_{R}} = \frac{w_i e(t)}{\sum_{i=1}^{R} w_i}$. Hence,

$$a_{R}(t+1) = a_{R}(t) + \Delta a_{R}(t) = a_{R}(t) + \eta_{11} S b_1 \frac{w_i e(t)}{\sum_{i=1}^{R} w_i}, \tag{20}$$

Therefore, we has update equation by b_{R} and c_{R} as

$$\Delta b_{R} = -\eta_{12} \frac{\partial \dot{V}}{\partial b_{R}} = -\eta_{12} \frac{\partial S \dot{S}}{\partial b_{R}} = -\eta_{12} S \frac{\partial \dot{S}}{\partial b_{R}}, \tag{21}$$

$$b_R(t+1) = b_R(t) + \Delta b_R(t) = b_R(t) + \eta_{12} S b_1 \frac{w_i \dot{e}(t)}{\sum_{i=1}^{R} w_i}, \tag{22}$$

where η_{12} is the learning rate, $\eta_{12} > 0$.

$$\Delta c_R = -\eta_{13} \frac{\partial \dot{V}}{\partial c_R} = -\eta_{13} \frac{\partial S\dot{S}}{\partial c_R} = -\eta_{13} S \frac{\partial \dot{S}}{\partial c_R}, \tag{23}$$

$$c_R(t+1) = c_R(t) + \Delta c_R(t) = c_R(t) + \eta_{13} S b_1 \frac{w_i}{\sum_{i=1}^{R} w_i}, \tag{24}$$

where η_{13} is the learning rate, $\eta_{13} > 0$.

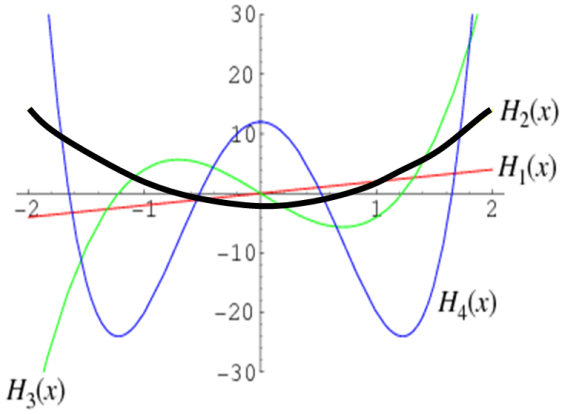

Fig. 4. Orthogonal Hermite polynomials

4.3. Robust Compensation with Adaptive Fuzzy Hermite Neural Networks

In neural networks applied to control systems, Gaussian functions are commonly employed. However, Gaussian functions have a drawback as they require parameters for their peak and width, necessitating more complex calculations to determine the optimal values for these parameters. In contrast, Hermite Polynomials have the advantage of expanding their input range with increasing order, eliminating the need for complex calculations to determine the optimal width. This not only simplifies the computational burden during system implementation but also reduces overall computational complexity.

Figure 4 displays the Orthogonal Hermite polynomials, with H_1 through H_4 representing polynomials of first to fourth order. Orthogonal Hermite polynomials exhibit a broader range compared to Gaussian functions. The paper proposes the Adaptive Fuzzy Hermite Neural Network (AFHNN), which incorporates Orthogonal Hermite polynomials, dynamic weight feedback, and robustness compensation. Finally, we employ Lyapunov stability to demonstrate system convergence. The AFHNN structure, depicted in

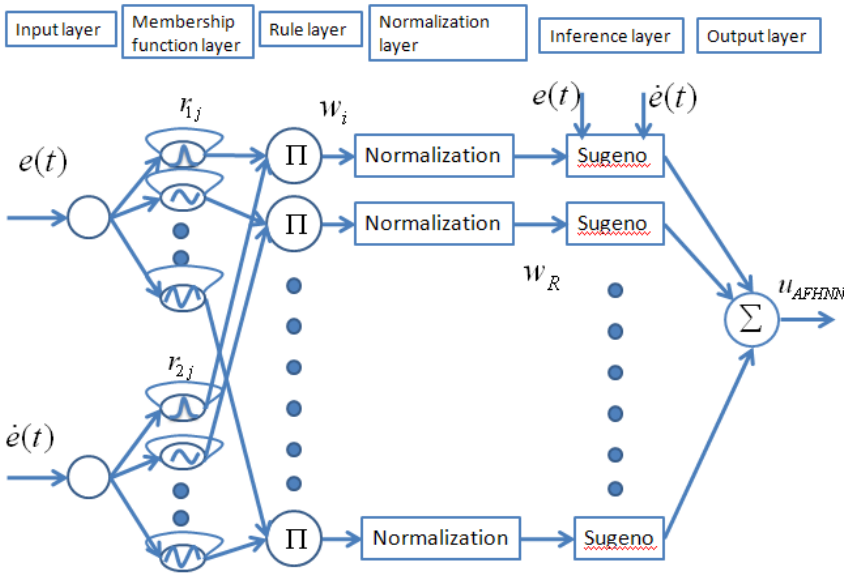

Fig. 5. The structure of AFHNN

Figure 5, consists of six layers. The first layer serves as the input layer, receiving external signals into the network. This can be expressed by the equation:

$$y_1^1 = e(t), \quad y_2^1 = \dot{e}(t), \tag{25}$$

where Superscript is the n-th network, and subscript is n-th input.

The second layer is membership function layer, in which is used fuzzification from first layer. The equation we can be expressed as

$$y_j^2 = \sigma_{j,k}(x) \tag{26}$$

$$y_{j+\max j}^2 = \sigma_{j+\max j,k}(x) \tag{27}$$

$$\sigma_{j,k}(x) = \frac{1}{\sqrt{2^j j! \sqrt{\pi}}} \exp^{-\vartheta_{j,k}^2/2} H_j(\vartheta_{j,k}) \tag{28}$$

$$\sigma_{j+\max j,k}(x) = \frac{1}{\sqrt{2^j j! \sqrt{\pi}}} \cdot \exp^{-\vartheta_{j+\max j,k}^2/2} \cdot H_{j+\max j}(\vartheta_{j+\max j,k}) \tag{29}$$

where j is the note. k is the simple time. $\vartheta_{j,k} = e(t) + r_{1j}\sigma_{j,k-1}$. exp is the exponential function. $\vartheta_{j+\max j,k} = \dot{e}(t) + r_{2j}\sigma_{j,k-1}$. $H_j(\vartheta_{j,k})$ is the Orthogonal Hermite polynomials. r_{1j} and r_{2j} is the recursive weight. $H_1(\vartheta_{j,k}) = 1, H_2(\vartheta_{j,k}) = 2\vartheta_{j,k}$, $H_n(\vartheta_{j,k}) = 2\vartheta_{j,k}H_{n-1}(\vartheta_{j,k}) - 2(n-1)H_{n-1}(\vartheta_{j,k})$ when $n \geq 3$.

The third layer is the rule layer, which is used logical product operator to second layer, so the output can be expressed as

$$y_i^3 = w_i = y_j^2 y_{j+\max j}^2, i = 1, 2, ..., Q \tag{30}$$

where Q is the rule number. The fourth layer is regularization layer, which is regulated to weight, and we can be expressed as

$$y_R^4 = w_R = \zeta_{R,k} = \frac{w_i}{\sum_{i=1}^{Q} w_i}, R = 1, 2, ..., Q \tag{31}$$

where k is the simple time of k-th. Fifth layer is the inference system, which is used Sugeno and average weighting method to defuzzification. The output can be expressed as

$$y_R^5 = w_R f_{AFHNN}(e(t), \dot{e}(t)) = \zeta_{R,k}(a_R e(t) + b_R \dot{e}(t) + c_R) = \zeta_{R,k} \varpi_R \tag{32}$$

where $a_R, b_R, c_R > 0, R = 1, 2, ..., Q$ is the inference function. The sixth layer is the output layer, which is used the linear combination of fifth layer, and the output can be expressed as

$$u_{AFHNN} = \sum_{R=1}^{n} y_R^5 = \mathbf{W^T}(\mathbf{A}, \mathbf{B}, \mathbf{C}) \cdot \boldsymbol{\varphi}(\mathbf{R_1}, \mathbf{R_2}) \tag{33}$$

where $\mathbf{W^T} = \left[\varpi_1, \ldots, \varpi_Q\right]_{1\times Q}$, $\boldsymbol{\varphi}^{\mathbf{T}} = \left[\zeta_1, \ldots, \zeta_Q\right]_{1\times Q}$, $\mathbf{A^T} = \left[a_1, \ldots, a_Q\right]_{1\times Q}$, $\mathbf{B^T} = \left[b_1, \ldots, b_Q\right]_{1\times Q}$, $\mathbf{C^T} = \left[c_1, \ldots, c_Q\right]_{1\times Q}$. $\mathbf{R_1^T} = \left[r_{11}, \ldots, r_{1j}\right]_{1\times j}$, $\mathbf{R_2^T} = \left[r_{21}, \ldots, r_{2j}\right]_{1\times j}$, $a_Q, b_Q, c_Q > 0$.

RCAFHNN used the Lyapunov function and feedback learning algorithms [24] to compensation output distribution. The control input define as

$$u = \frac{-1}{b_1(x)}\Big(-\dot{x}_1 + f_1(x) + E(x) - h_1\ddot{e}(t) + \dot{e}(t) + h_2 e(t) + h_3 \int_0^t e(t)dt\Big) = \hat{u} + \varepsilon_1 \tag{34}$$

where $\hat{u}$ is the output of RCAFHNN, ε_1 is the error between u and $\hat{u}$. In the formula of equation 34, the SRM parameters and lumped uncertainty are unknown. Therefore, we use AFHNN to track u. Substituting equation 34 to equation 8 can be obtained

$$\dot{e}(t) = -h_1\ddot{e}(t) + h_2 e(t) + h_3 \int_0^t e(t)dt + (u - \hat{u} - \varepsilon_1) \tag{35}$$

where $u - \hat{u} - \varepsilon_1 = 0$

Define the estimate error of AFHNN as

$$\begin{aligned} \widetilde{u} &= u - \hat{u} = \mathbf{W}^{*\mathbf{T}}(\mathbf{A}^*, \mathbf{B}^*, \mathbf{C}^*)\,\phi^*(\mathbf{R}_1^*, \mathbf{R}_2^*) - \hat{\mathbf{W}}^{\mathrm{T}}(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}})\hat{\boldsymbol{\varphi}}\left(\hat{\mathbf{R}}_1, \hat{\mathbf{R}}_2\right) - u_{ss} \\ &= \mathbf{W}^{*\,\mathrm{T}}\widetilde{\boldsymbol{\varphi}} + \widetilde{\mathbf{W}}^{\mathrm{T}}\hat{\boldsymbol{\varphi}} - u_{ss} \end{aligned} \tag{36}$$

where $\hat{u} = u_{AFHNN} + u_{ss}, u_{ss}$ is the control output of robustness compensation. $\widetilde{\mathbf{W}} = \mathbf{W}^* - \hat{\mathbf{W}}, \widetilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}^* - \hat{\boldsymbol{\varphi}}, \mathbf{A}^*, \mathbf{B}^*, \mathbf{C}^*$ are the approximation weight of default control input. $\mathbf{R}_1^*, \mathbf{R}_2^*$ are the approximation recursive weight of default control input. $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}$ are the weight of AFHNN. $\hat{\mathbf{R}}_1, \hat{\mathbf{R}}_2$ are the recursive weight of AFHNN.

Define as

$$\begin{aligned}\tilde{\mathbf{W}} = \begin{bmatrix}\widetilde{\varpi}_1 \\ \vdots \\ \widetilde{\varpi}_Q\end{bmatrix} = \left.\begin{bmatrix}\frac{\partial \varpi_1}{\partial \mathbf{A}^{\mathbf{T}}} \\ \vdots \\ \frac{\partial \bar{\varpi}_Q}{\partial \mathbf{A}^{\mathbf{T}}}\end{bmatrix}\right|_{\mathbf{A}=\hat{\mathbf{A}}} \left(\mathbf{A}^* - \hat{\mathbf{A}}\right) + \left.\begin{bmatrix}\frac{\partial \varpi_1}{\partial \mathbf{B}^{\mathbf{T}}} \\ \vdots \\ \frac{\partial \bar{\varpi}_Q}{\partial \mathbf{B}^{\mathbf{T}}}\end{bmatrix}\right|_{\mathbf{B}=\hat{\mathbf{B}}} \left(\mathbf{B}^* - \hat{\mathbf{B}}\right) \\ + \left.\begin{bmatrix}\frac{\partial \varpi_1}{\partial \mathbf{C}^{\mathbf{T}}} \\ \vdots \\ \frac{\partial \bar{\varpi}_Q}{\partial \mathbf{C}^{\mathbf{T}}}\end{bmatrix}\right|_{\mathbf{C}=\hat{\mathbf{C}}} \left(\mathbf{C}^* - \hat{\mathbf{C}}\right) + \boldsymbol{\varphi}_{\mathbf{H2}} = \mathbf{W}_{\mathrm{A}}^{\mathrm{T}}\widetilde{\mathbf{A}} + \mathbf{W}_{\mathrm{B}}^{\mathrm{T}}\widetilde{\mathbf{B}} + \mathbf{W}_{\mathrm{C}}^{\mathrm{T}}\widetilde{\mathbf{C}} + \varphi_{\mathrm{H2}}\end{aligned} \tag{37}$$

$$\begin{aligned}\widetilde{\boldsymbol{\varphi}} = \begin{bmatrix}\widetilde{\varsigma}_1 \\ \vdots \\ \widetilde{\varsigma}_Q\end{bmatrix} = \left.\begin{bmatrix}\frac{1}{2}\left(\frac{\partial \zeta_1}{\partial \mathbf{R}_1^{\mathrm{T}}} + \frac{\partial \zeta_1}{\partial \zeta_{j,k-1}}\frac{\partial \zeta_{j,k-1}}{\partial \mathbf{R}_1^{\mathrm{T}}}\right) \\ \vdots \\ \frac{1}{2}\left(\frac{\partial \zeta_Q}{\partial \mathbf{R}_1^{\mathrm{T}}} + \frac{\partial \zeta_Q}{\partial \zeta_{Q,k-1}}\frac{\partial \zeta_{Q,k-1}}{\partial \mathbf{R}_1^{\mathrm{T}}}\right)\end{bmatrix}\right|_{\mathbf{R}_1-\hat{\mathbf{R}}_1} \left(\mathbf{R}_1^* - \hat{\mathbf{R}}_1\right) \\ + \left.\begin{bmatrix}\frac{1}{2}\left(\frac{\partial \zeta_1}{\partial \mathbf{R}_2^{\mathrm{T}}} + \frac{\partial \zeta_1}{\partial \zeta_{\mathrm{j},k-1}}\frac{\partial \zeta_{j,k-1}}{\partial \mathbf{R}_2^{\mathrm{T}}}\right) \\ \vdots \\ \frac{1}{2}\left(\frac{\partial \zeta_Q}{\partial \mathbf{R}_2^{\mathrm{T}}} + \frac{\partial \zeta_Q}{\partial \zeta_{\mathrm{Q},k-1}}\frac{\partial \zeta_{Q,k-1}}{\partial \mathbf{R}_2^{\mathrm{T}}}\right)\end{bmatrix}\right|_{\mathbf{R}_2-\hat{\mathbf{R}}_2} \left(\mathbf{R}_2^* - \hat{\mathbf{R}}_2\right) + \boldsymbol{\varphi}_{\mathrm{H1}} \\ = \varphi_{\mathrm{R}_1}^{\mathrm{T}}\widetilde{\mathbf{R}}_1 + \varphi_{\mathrm{R}_2}^{\mathrm{T}}\widetilde{\mathbf{R}}_2 + \varphi_{\mathrm{Hl}}\end{aligned} \tag{38}$$

where

$$\mathbf{W}_{\mathbf{A}} = \left.\begin{bmatrix}\frac{\partial \varpi_1}{\partial a_1} & \frac{\partial \varpi_2}{\partial a_1} & \cdots & \frac{\partial \varpi_Q}{\partial a} \\ \frac{\partial \varpi_1}{\partial a_2} & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \varpi_1}{\partial a_Q} & \frac{\partial \varpi_2}{\partial a_Q} & \cdots & \frac{\partial \varpi_Q}{\partial a_Q}\end{bmatrix}_{Q\times Q}\right|_{A=\hat{A}};$$

$$\mathbf{W}_{\mathbf{B}} = \left.\begin{bmatrix}\frac{\partial \varpi_1}{\partial \mathbf{B}} & \frac{\partial \varpi_2}{\partial \mathbf{B}} & \cdots & \frac{\partial \varpi_Q}{\partial \mathbf{B}}\end{bmatrix}_{Q\times Q}\right|_{\mathbf{B}=\hat{\mathbf{B}}};$$

$$\mathbf{W}_{\mathbf{C}} = \left.\begin{bmatrix}\frac{\partial \varpi_1}{\partial \mathbf{C}} & \frac{\partial \varpi_2}{\partial \mathbf{C}} & \cdots & \frac{\partial \varpi_Q}{\partial \mathbf{C}}\end{bmatrix}_{Q\times Q}\right|_{\mathbf{C}=\hat{\mathbf{C}}};$$

$$\varphi_{\mathbf{R_1^T}} = \left[\begin{array}{c} \frac{1}{2}\left(\frac{\partial \zeta_1}{\partial \mathbf{R_1}} + \frac{\partial \zeta_1}{\partial \zeta_{1,k-1}} \frac{\partial \zeta_{1,k-1}}{\partial \mathbf{R_1}}\right) \\ \frac{1}{2}\left(\frac{\partial \zeta_2}{\partial \mathbf{R_1}} + \frac{\partial \zeta_2}{\partial \zeta_{2,k-1}} \frac{\partial \zeta_{2,k-1}}{\partial \mathbf{R_1}}\right) \\ \vdots \\ \frac{1}{2}\left(\frac{\partial \zeta_Q}{\partial \mathbf{R_1}} + \frac{\partial \zeta_Q}{\partial \zeta_{Q,k-1}} \frac{\partial \zeta_{Q,k-1}}{\partial \mathbf{R_1}}\right) \end{array} \right]_{Q \times j} \Bigg|_{R_1 = \hat{R}_1} ;$$

$$\varphi_{\mathbf{R_2^T}} = \left[\begin{array}{c} \frac{1}{2}\left(\frac{\partial \zeta_1}{\partial \mathbf{R_2}} + \frac{\partial \zeta_1}{\partial \zeta_{1,k-1}} \frac{\partial \zeta_{1,k-1}}{\partial \mathbf{R_1}}\right) \\ \frac{1}{2}\left(\frac{\partial \zeta_2}{\partial \mathbf{R_2}} + \frac{\partial \zeta_2}{\partial \zeta_{2,k-1}} \frac{\partial \zeta_{2,k-1}}{\partial \mathbf{R_1}}\right) \\ \vdots \\ \frac{1}{2}\left(\frac{\partial \zeta_Q}{\partial \mathbf{R_2}} + \frac{\partial \zeta_Q}{\partial \zeta_{Q,k-1}} \frac{\partial \zeta_{Q,k-1}}{\partial \mathbf{R_1}}\right) \end{array} \right]_{Q \times j} \Bigg|_{R_2 = \hat{R}_2} ;$$

$$\widetilde{\mathbf{A}} = \mathbf{A}^* - \hat{\mathbf{A}}; \widetilde{\mathbf{B}} = \mathbf{B}^* - \hat{\mathbf{B}}; \widetilde{\mathbf{C}} = \mathbf{C}^* - \hat{\mathbf{C}}; \widetilde{\mathbf{R}}_1 = \mathbf{R}_1^* - \hat{\mathbf{R}}_1; \widetilde{\mathbf{R}}_2 = \mathbf{R}_2^* - \hat{\mathbf{R}}_2$$

$\varphi_{\mathrm{H1}}, \varphi_{\mathrm{H2}}$ are the higher-order error in Taylor expansion.

Substituting equation 37-38 to equation 36 can be obtained

$$\begin{aligned} \widetilde{u} &= \mathbf{W}^{\star \mathrm{T}} \widetilde{\boldsymbol{\varphi}} + \widetilde{\mathbf{W}}^{\mathrm{T}} \hat{\boldsymbol{\varphi}} - u_{ss} = \hat{\mathbf{W}}^{\mathrm{T}} \widetilde{\boldsymbol{\varphi}} + \widetilde{\mathbf{W}}^{\mathrm{T}} \widetilde{\boldsymbol{\varphi}} + \widetilde{\mathbf{W}}^{\mathrm{T}} \hat{\boldsymbol{\varphi}} - u_{ss} \\ &= \hat{\mathbf{W}}^{\mathrm{T}} \left(\boldsymbol{\varphi}_{\mathrm{R}_1}^{\mathrm{T}} \widetilde{\mathbf{R}}_1 + \boldsymbol{\varphi}_{\mathrm{R}_2}^{\mathrm{T}} \widetilde{\mathbf{R}}_2 \right) + \left(\mathbf{W}_{\mathrm{A}}^{\mathrm{T}} \widetilde{\mathbf{A}} + \mathbf{W}_{\mathbf{B}}^{\mathrm{T}} \widetilde{\mathbf{B}} + \mathbf{W}_{\mathrm{C}}^{\mathrm{T}} \widetilde{\mathbf{C}} \right)^{\mathrm{T}} \hat{\boldsymbol{\varphi}} - u_{ss} + L_1 \end{aligned} \tag{39}$$

where $L_1 = \widetilde{\mathbf{W}}^{\mathrm{T}} \widetilde{\boldsymbol{\varphi}} + \boldsymbol{\varphi}_{\mathrm{H2}}^{\mathrm{T}} \hat{\boldsymbol{\varphi}} + \hat{\mathbf{W}}^{\mathrm{T}} \varphi_{\mathrm{Hl}}$ is the total estimation error in AFHNN. Define Lyapunov function as

$$\begin{aligned} V_2 = \frac{1}{2} S^2 &+ \frac{1}{2n_1} \widetilde{\mathbf{A}}^{\mathrm{T}} \widetilde{\mathbf{A}} + \frac{1}{2n_2} \widetilde{\mathbf{B}}^{\mathrm{T}} \widetilde{\mathbf{B}} + \frac{1}{2n_3} \widetilde{\mathbf{C}}^{\mathrm{T}} \widetilde{\mathbf{C}} \\ &+ \frac{1}{2n_4} \widetilde{\mathbf{R}}_1^{\mathrm{T}} \widetilde{\mathbf{R}}_1 + \frac{1}{2n_5} \widetilde{\mathbf{R}}_2^{\mathrm{T}} \widetilde{\mathbf{R}}_2 + \frac{1}{2n_6} \widetilde{L}^2 \end{aligned} \tag{40}$$

where $\widetilde{L} = L - \hat{L}$, L is the lump uncertainty of RCAFHNN and system, $S(t) = h_1 \dot{e}(t) + e(t), \quad n_1, n_2, n_3, n_4, n_5, n_6 > 0$.

Differential the equation (38), and subsisting equation (37), we get

$$\begin{aligned}
\dot{V}_2 &= S\dot{S} - \frac{1}{n_1}\widetilde{\mathbf{A}}^{\mathrm{T}}\dot{\hat{\mathbf{A}}} - \frac{1}{n_2}\widetilde{\mathbf{B}}^{\mathrm{T}}\dot{\hat{\mathbf{B}}} - \frac{1}{n_3}\widetilde{\mathbf{C}}^{\mathrm{T}}\dot{\hat{\mathbf{C}}} - \frac{1}{n_4}\widetilde{\mathbf{R}}_1^{\mathrm{T}}\dot{\hat{\mathbf{R}}}_1 - \frac{1}{n_5}\widetilde{\mathbf{R}}_2^{\mathrm{T}}\dot{\hat{\mathbf{R}}}_2 - \frac{1}{n_6}\widetilde{\hat{L}}\dot{\hat{L}} \\
&= S\left(-h_1\ddot{e}(t) + h_2 e(t) + h_3\int_0^t e(t)dt + (u - \hat{u} - \varepsilon_1) + h_1\ddot{e}\right) - \frac{1}{n_1}\widetilde{\mathbf{A}}^{\mathrm{T}}\dot{\hat{\mathbf{A}}} - \frac{1}{n_2}\widetilde{\mathbf{B}}^{\mathrm{T}}\dot{\hat{\mathbf{B}}} \\
&- \frac{1}{n_3}\widetilde{\mathbf{C}}^{\mathrm{T}}\dot{\hat{\mathbf{C}}} - \frac{1}{n_4}\widetilde{\mathbf{R}}_1^{\mathrm{T}}\dot{\hat{\mathbf{R}}}_{\mathbf{1}} - \frac{1}{n_5}\widetilde{\mathbf{R}}_2^{\mathrm{T}}\dot{\hat{\mathbf{R}}}_2 - \frac{1}{n_6}\tilde{\hat{L}}\dot{\hat{L}} \\
&= S\left(h_2 e(t) + h_3\int_0^t e(t)dt + \widetilde{u} - \varepsilon_1\right) - \frac{1}{n_1}\widetilde{\mathbf{A}}^{\mathrm{T}}\dot{\hat{\mathbf{A}}} - \frac{1}{n_2}\widetilde{\mathbf{B}}^{\mathrm{T}}\dot{\hat{\mathbf{B}}} - \frac{1}{n_3}\widetilde{\mathbf{C}}^{\mathrm{T}}\dot{\hat{\mathbf{C}}} - \frac{1}{n_4}\widetilde{\mathbf{R}}_1^{\mathrm{T}}\dot{\hat{\mathbf{R}}}_1 \\
&- \frac{1}{n_5}\widetilde{\mathbf{R}}_2^{\mathrm{T}}\dot{\hat{\mathbf{R}}}2 - \frac{1}{n_6}\widetilde{\hat{L}}\dot{\hat{L}} \\
&= S\Big(\varepsilon_1 + \hat{\mathbf{W}}^{\mathrm{T}}\left(\boldsymbol{\varphi}_{\mathbf{R}_1}^{\mathrm{T}}\widetilde{\mathbf{R}}_1 + \boldsymbol{\varphi}_{\mathbf{R}_2}^{\mathrm{T}}\widetilde{\mathbf{R}}_2\right) + \left(\mathbf{W}_{\mathbf{A}}^{\mathrm{T}}\widetilde{\mathbf{A}} + \mathbf{W}_{\mathbf{B}}^{\mathrm{T}}\widetilde{\mathbf{B}} + \mathbf{W}_{\mathbf{C}}^{\mathrm{T}}\widetilde{\mathbf{C}}\right)\hat{\boldsymbol{\varphi}} - u_{ss} + h_2 e(t) \\
&+ h_3\int_0^t e(t)dt + L_1 - \varepsilon_1\Big) - \frac{1}{n_1}\widetilde{\mathbf{A}}^{\mathrm{T}}\dot{\hat{\mathbf{A}}} - \frac{1}{n_2}\widetilde{\mathbf{B}}^{\mathrm{T}}\dot{\hat{\mathbf{B}}} - \frac{1}{n_3}\widetilde{\mathbf{C}}^{\mathrm{T}}\dot{\hat{\mathbf{C}}} - \frac{1}{n_4}\widetilde{\mathbf{R}}_1^{\mathrm{T}}\dot{\hat{\mathbf{R}}}_1 - \frac{1}{n_5}\widetilde{\mathbf{R}}_2^{\mathrm{T}}\dot{\hat{\mathbf{R}}}_2 \\
&- \frac{1}{n_6}\tilde{\hat{L}}\dot{\hat{L}}
\end{aligned} \tag{41}$$

Define as

$$L = (-\varepsilon_1 + L_1) \tag{42}$$

Therefore, we can get the adaptive law and robust compensation as

$$u_{ss} = h_2 e + h_3\int_0^t e(t)dt + k_v S + \hat{L} \tag{43}$$

$$\dot{\hat{\mathbf{A}}} = n_1 S\mathbf{W}_{\mathbf{A}}\hat{\boldsymbol{\varphi}} \tag{44}$$

$$\dot{\hat{\mathbf{B}}} = n_2 S\mathbf{W}_{\mathbf{B}}\hat{\boldsymbol{\varphi}} \tag{45}$$

$$\dot{\hat{\mathbf{C}}} = n_3 S\mathbf{W}_{\mathbf{C}}\hat{\boldsymbol{\varphi}} \tag{46}$$

$$\dot{\hat{\mathbf{R}}}_1 = n_4 S\boldsymbol{\varphi}_{R_1}\hat{\mathbf{W}} \tag{47}$$

$$\dot{\hat{\mathbf{R}}}_2 = n_5 S\boldsymbol{\varphi}_{R_2}\hat{\mathbf{W}} \tag{48}$$

$$\dot{\hat{L}} = n_6 S \tag{49}$$

As we can observe in equations (43) to (49), the input control variables used do not depend on system parameters. In other words, the proposed controller in this paper can be applied to parameterless systems as well as nonlinear systems. The use of Lyapunov convergence criteria ensures the updating of neural network parameters, overcoming uncertainties during the operation of the motor system. Replacing traditional Gaussian functions with Hermite Polynomials eliminates the need to calculate optimal peak and width parameters. Substituting equation 43-49 to 41, we have

$$\dot{V} = -k_v S^2 \leq 0 \tag{50}$$

We can know the SRM is convergence of Lyapunov function by 50. Then define as

$$\xi(t) = k_v S^2 \tag{51}$$

Integrating equation 51, we have

$$\int_0^t \xi(\tau)d\tau = V\Big(S(0)\Big) - V\Big(S(t)\Big) \tag{52}$$

Because $V\Big(S(0)\Big)$ and $V\Big(S(t)\Big)$ are bounded, hence

$$\lim_{t\to\infty} \int_0^t \xi(\tau)d\tau < \infty \tag{53}$$

According Barbalat lemma [17], we have

$$\lim_{t\to\infty} \xi(\tau) = 0 \tag{54}$$

When $t \to \infty$, then $S \to 0$ and height error $e(t) \to 0$.

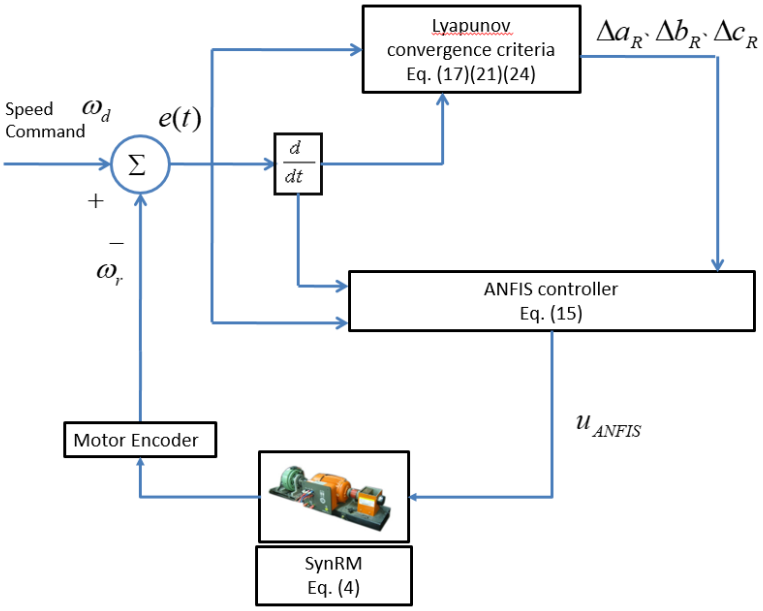

Fig. 6. Block of ANFIS

5. Experimental results

In the experiments, we aim to compare the differences between using ANFIS and the proposed neural control method in SRMs (Synchronous Reluctance Motors).

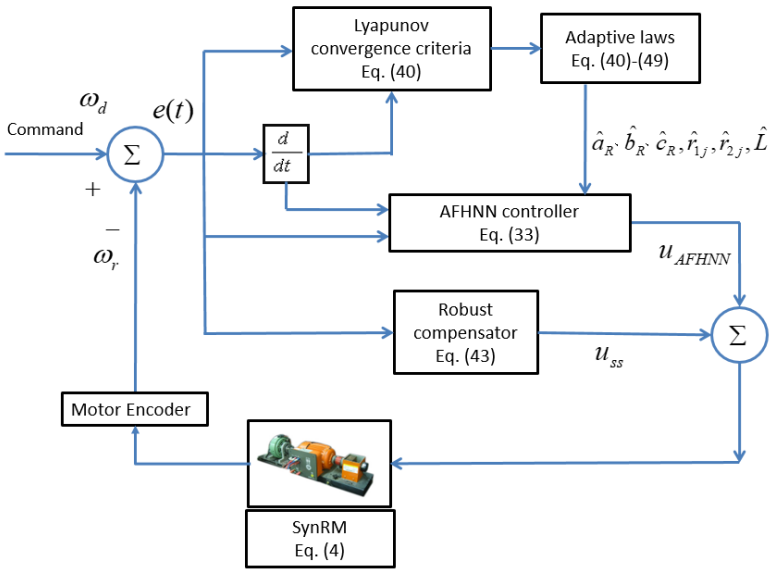

Fig. 7. Block of RCAFHNN

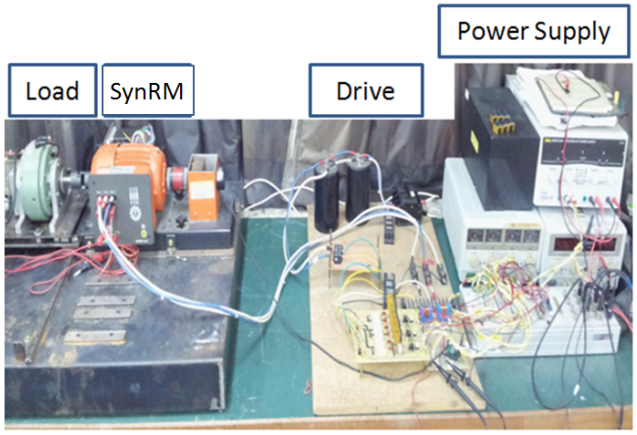

Fig. 8. Synchronous reluctance motors equipment

We have designed experiments to track motor speed errors in various demanding control scenarios during experimental testing. These scenarios include motor operation under no-load conditions, loaded conditions, and with different speed commands. We will assess the performance of the velocity controller in response to these scenarios.

The ANFIS work environment is illustrated in Figure 6. Initially, the command speed is set using a computer, and the system calculates the error between the command speed and the system output. The error signal is then fed into ANFIS, and the control input is calculated. Finally, the Lyapunov function is utilized to adjust the ANFIS weight values until the error approaches zero.

The RCAFHNN work environment is depicted in Figure 7. Similarly, the command speed is set using a computer, and the system calculates the error between the command speed and the system output. This process yields both the error and differential error signals. These signals are then input into AFHNN, and the control output is calculated to yield u_{AFHNN} and u_{ss}. Finally, the Lyapunov function is employed to adjust the AFHNN weight values until the error approaches zero, and the robust composition controller compensates for the lump uncertainty of SRM.

The RCAFHNN demonstrates an improvement in handling lump uncertainty, parameter variations, and external load in SRMs. Figure 8 illustrates the experimental SRM equipment. The controller was implemented using the ds1104 Card from dSPACE Company. The parameters utilized in this study are presented in Table 1.

Table 1. controller and SynRM parameters

Control Methods	Motor Parameters	Controller Parameters	Public Parameters
ANFIS	$J_m = 0.00076$	$\eta_1 = 0.01, \quad \eta_2 = 0.01$	$a_{1\sim9} = 2, \quad b_{1\sim9} = 50$
	$B_m = 0.00012$	$\eta_3 = 0.01$	$c_1 = -0.1, \quad c_2 = -0.1$
RCAFHNN		$h_2 = 60, \quad h_3 = 1$	$c_3 = 0, \quad c_4 = -0.1$
		$k_v = 100 \quad n_1 = 0.01$	$c_5 = 0, \quad c_6 = 0.1$
		$n_2 = 0.01 \quad n_3 = 0.001$	$c_7 = 0, \quad c_8 = 0.1$
		$n_4 = 0.001 \quad n_5 = 0.001$	$c_9 = 0.1, \quad j = 3$
		$n_6 = 0.001 \quad n_7 = 0.001$	$h_1 = 90, \quad R = 9$
		$n_8 = 20$	

Figure 9 shows the simulation output responses, error responses, A phase current comparison for initial command speed 800rpm at $0 \leq t < 5$ sec, and the changed command speed 1200rpm at $t \geq 5$ sec of ANFIS and RCAFHNN. In Figure 9, RCAFHNN can track command speed faster than ANFIS at transient response, and accurate steady-state tracking speed when the command speed is changed.

Figure 10 are the simulation output responses, error responses, A phase current comparison for time varying command speed $800+100\sin(2\pi t)$ rpm of ANFIS and RCAFHNN. In Figure 10, RCAFHNN has better tracking ability and error faster convergent.

Figure 11 is shown command speed 600rpm and initial external load is added 0.35NT-m, then we change external load is added 0.9NT-m at $t \geq 10$ of ANFIS and RCAFHNN. In figure 11, we show output response, output amplifier response, error response, A phase current comparison, neural network output and Phase plane for the error and differen-

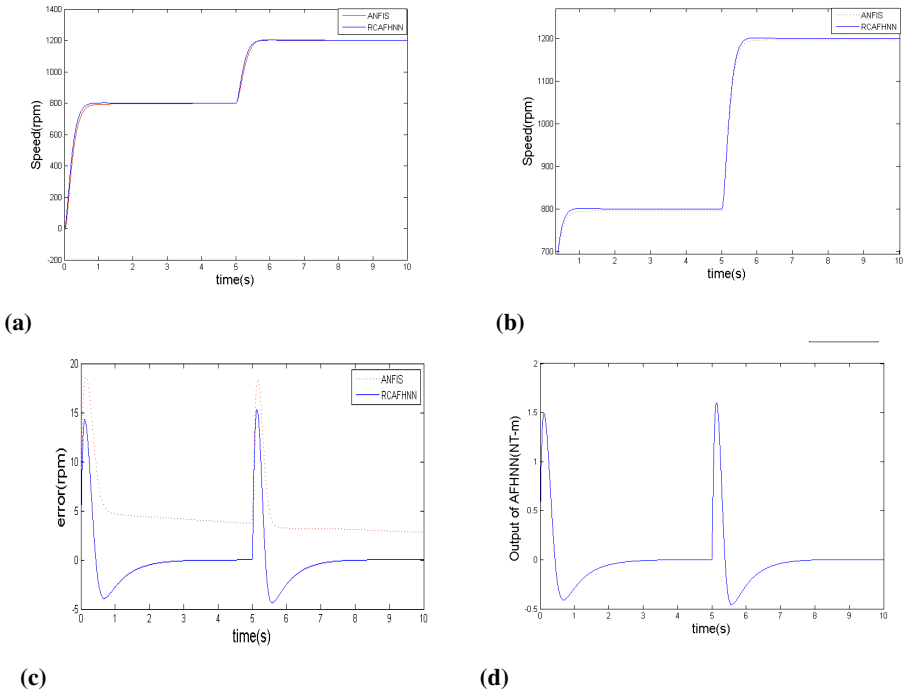

Fig. 9. Simulation responses of RCAFHNN and ANFIS for command speed 800rpm at $0 \leq t < 5$ and 1200rpm speed command at $t \geq 5$ sec (a) comparison of output responses, (b) zoomed-in comparison of output responses, (c) comparison of error responses, (d) output of AFHNN, (e) robust compensation

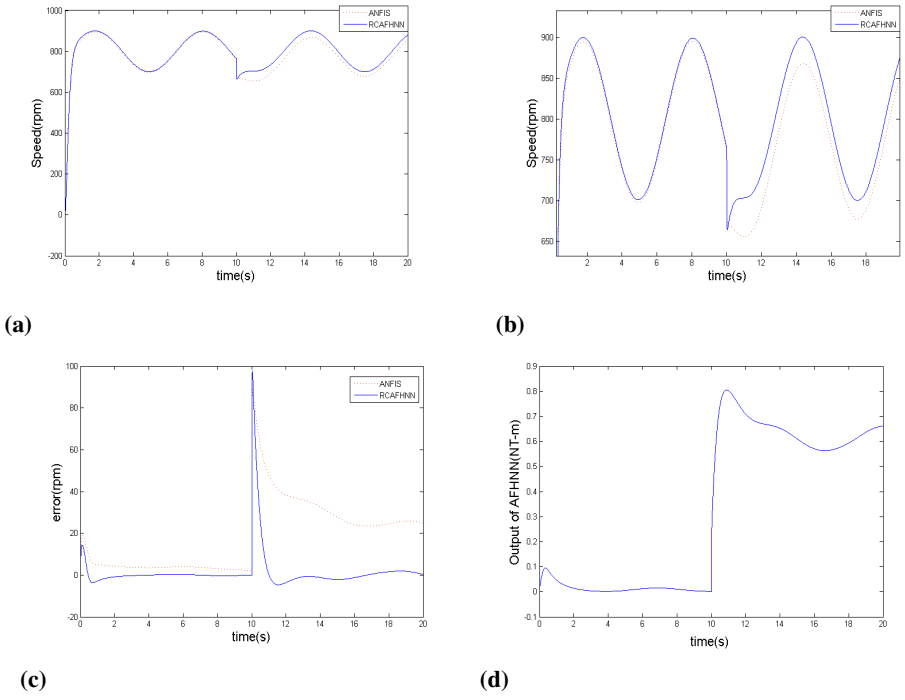

Fig. 10. Simulation responses of RCAFHNN and ANFIS at command speed $800 + 100\sin(2\pi t)$ and added external load 0.8 NT-m at $t \geq 10$ seconds (a) comparison of output responses, (b) zoomed-in comparison of output responses, (c) comparison of error responses, (d) output of AFHNN

tial error. Figure 11 (a)-(d), ANFIS tracking slowly of command speed at transient state. RCAFHNN has the faster tracking error and stability control output.

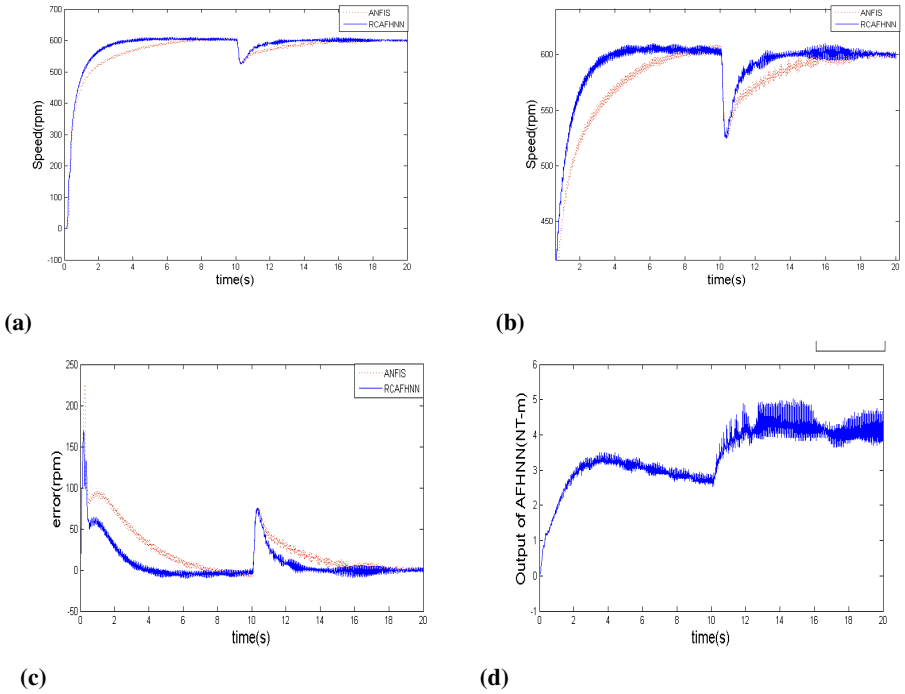

Fig. 11. Experimental responses of RCAFHNN and ANFIS at command speed 600rpm an 0.35NT-m external load is added at initial. At $t \geq 10$ seconds, an 0.9NT-m external load is added. (a) comparison of output responses, (b) zoomed-in comparison of output responses, (c) comparison of error responses, (d) Output of AFHNN

Figure 12 is shown command speed 600rpm and initial external load is added 0.35NT-m, then we change command speed 800rpm at $t \geq 5$ and external load is added 0.9NT-m at $t \geq 10$ of ANFIS and RCAFHNN. In figure 12, we show output response, output amplifier response and error response, A phase current comparison and neural network output. In figure 12, we can know that RCAFHNN has better tracking error when change the command speed and external load.

Figure 13 is shown time varies command speed $700+100\sin(2\pi t)$ rpm and initial external load is added 0.35NT-m, then we change external load is added 0.9NT-m at $t \geq 10$ of ANFIS and RCAFHNN. In figure 13, we show output response, output amplifier response, error response, A phase current comparison, and neural network output. In figure 13, RCAFHNN track the sine wave has the better ability than ANFIS, and RCAFHNN has faster track error when change external load.

Table 2 and Table 3 compares the experimental RMSEs. The performance index, RMSE, is defined as follows:

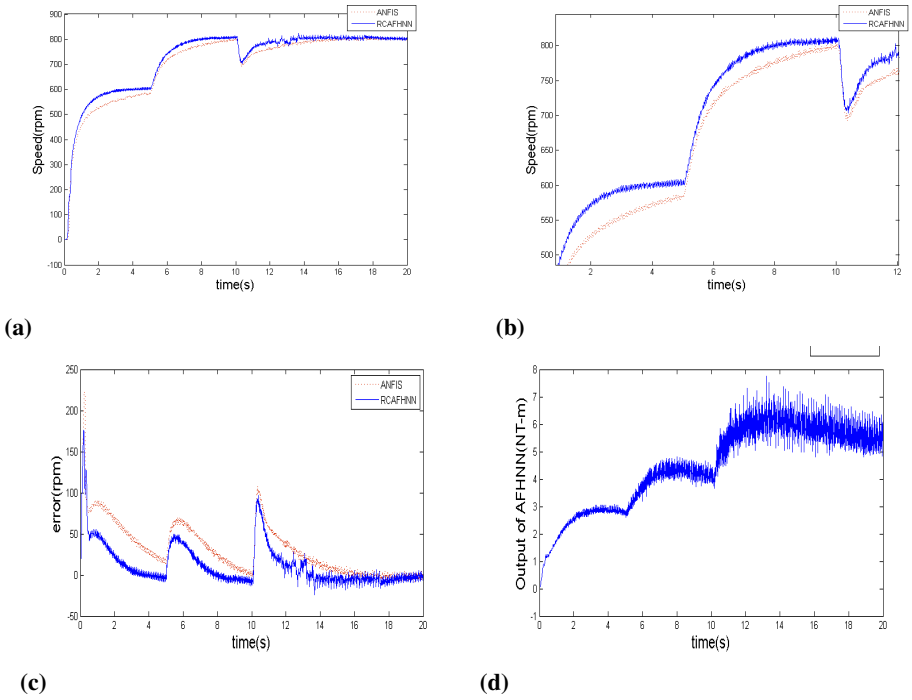

Fig. 12. Experimental responses of RCAFHNN and ANFIS at command speed 600rpm is $0 \leq t < 5$ seconds and 800rpm speed command in with an 0.35NT-m external load is added at initial. At $t \geq 10$ seconds, an 0.9NT-m external load is added. (a) comparison of output responses, (b) zoomed-in comparison of output responses, (c) comparison of error responses, (d) Output of AFHNN

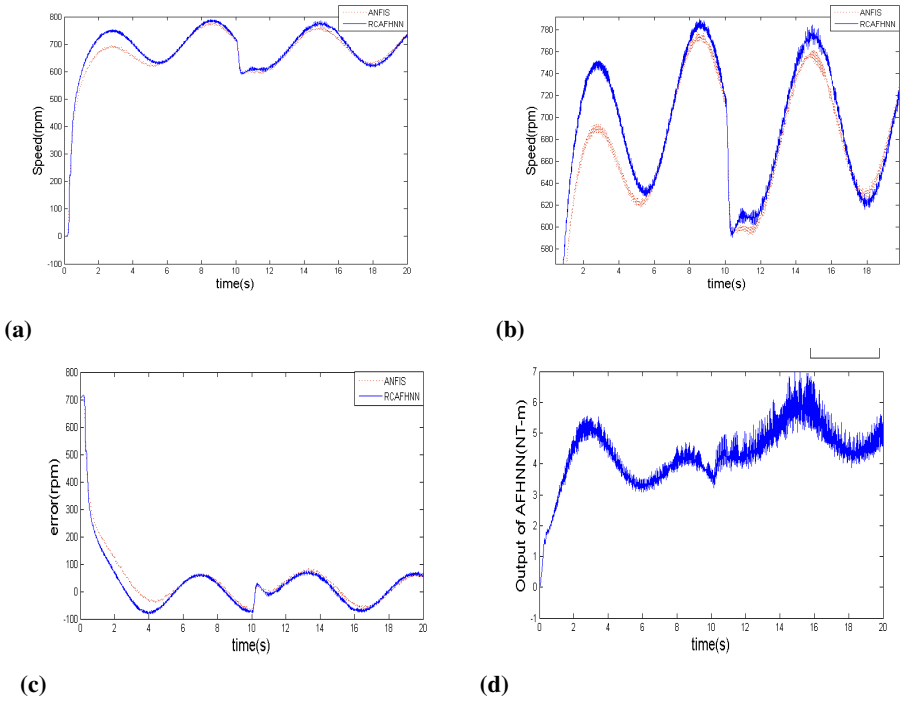

Fig. 13. Experimental responses of RCAFHNN and ANFIS at command speed $700 + 100\sin(2\pi t)$ rpm and an 0.35NT-m external load is added at initial. At $t \geq 10$ seconds, an 0.9NT-m external load is added. (a) comparison of output responses, (b) zoomed-in comparison of output responses, (c) comparison of error responses, (d) Output of AFHNN

$$RMSE = \sqrt{\frac{\sum_{i=1}^{\alpha} e^2\,[i]}{\alpha}} \tag{55}$$

where α is the number of the sampled points. Table 2 and Table 3 clearly demonstrates that RCAFHNN outperforms the ANFIS schemes under all operational conditions because of its energy control input is consider in controller. The experimental results conclusively establish the regulation ability of the proposed RCAFHNN over a wide range of speeds, its dynamic tracking capability, and its robustness.

Table 2. Simulation Comparsion of RMSE

Control Methods	800rpm to 1200rpm	800+100sin$(2\pi t)$ rpm with load
ANFIS	0.3268	1.5906
RCAFHNN	0.1831	1.2920

Table 3. Implement Comparsion of RMSE with load

Control Methods	600rpm	600rpm to 800 rpm	700+100 sin$(2\pi t)$rpm
ANFIS	40.0064	44.9906	120.2712
RCAFHNN	25.4595	28.2062	117.5688

6. Conclusion

This study successfully implemented the RCAFHNN (robust compensation scheme using adaptive fuzzy Hermite neural networks) in an SRM (synchronous reluctance motor). The RCAFHNN used adaptive laws to train weights online. Lyapunov stability was used to confirm the stability of the SRM. Moreover, the RCAFHNN offered satisfactory performance in handling lumped uncertainty and nonlinear dynamics. Finally, it can adapt to and track changes in speed and external load at transient and steady states, in spite of sine waves. Simulation and experimental results demonstrated the advantages of the proposed method.

References

1. Abootorabi Zarchi, H., Soltani, J., Arab Markadeh, G.: Adaptive input–output feedback-linearization-based torque control of synchronous reluctance motor without mechanical sensor. IEEE Transactions on Industrial Electronics 57(1), 375–384 (2010)
2. Barcaro, M., Bianchi, N., Magnussen, F.: Permanent-magnet optimization in permanent-magnet-assisted synchronous reluctance motor for a wide constant-power speed range. IEEE Transactions on Industrial Electronics 59(6), 2495–2502 (2012)

3. Castaneda, C.E., Loukianov, A.G., Sanchez, E.N., Castillo-Toledo, B.: Discrete-time neural sliding-mode block control for a dc motor with controlled flux. IEEE Transactions on Industrial Electronics 59(2), 1194–1207 (2012)
4. Cavanough, G.L., Kochanek, M., Cunningham, J.B., Gipps, I.D.: A self-optimizing control system for hard rock percussive drilling. IEEE/ASME Transactions on Mechatronics 13(2), 153–157 (2008)
5. Chen, C.S., Chen, J.S.: Rotor fault diagnosis system based on sga-based individual neural networks. Expert Systems with Applications 38(9), 10822–10830 (2011)
6. Choi, H.H., Jung, J.W., Kim, R.Y.: Fuzzy adaptive speed control of a permanent magnet synchronous motor. International Journal of Electronics 35(6) (2012)
7. El-Sousy, F.F.M.: Robust wavelet-neural-network sliding-mode control system for permanent magnet synchronous motor drive. IET Electric Power Applications 5(1), 113–132 (2011)
8. Ghaderi, A., Hanamoto, T.: Wide-speed-range sensorless vector control of synchronous reluctance motors based on extended programmable cascaded low-pass filters. IEEE Transactions on Industrial Electronics 58(6), 2322–2333 (2011)
9. Gong, W., Liang, J., Kan, X., Nie, X.: Robust state estimation for delayed complex-valued neural networks. Neural Processing Letters 46, 1009–1029 (2017)
10. Hsiao, F.H.: Neural-network based approach on delay-dependent robust stability criteria for dithered chaotic systems with multiple time-delay. Neurocomputing 191, 161–174 (2016)
11. Huang, H.C., Chiang, C.H.: An evolutionary radial basis function neural network with robust genetic-based immunecomputing for online tracking control of autonomous robots. Neural Process. Lett. 44(1), 19–35 (aug 2016)
12. Jon, R., Wang, Z., Luo, C., Jong, M.: Adaptive robust speed control based on recurrent elman neural network for sensorless pmsm servo drives. Neurocomputing 227, 131–141 (2017)
13. Kazemy, A., Hosseini, S.A., Farrokhi, M.: Second order diagonal recurrent neural network. In: 2007 IEEE International Symposium on Industrial Electronics. pp. 251–256 (2007)
14. Liu, M., Dong, M., Wu, C.: A new anfis for parameter prediction with numeric and categorical inputs. IEEE Transactions on Automation Science and Engineering 7(3), 645–653 (2010)
15. Ma, L., Khorasani, K.: Constructive feedforward neural networks using hermite polynomial activation functions. IEEE Transactions on Neural Networks 16(4), 821–833 (2005)
16. Nam, K.T., Kim, H., Lee, S.J., Kuc, T.Y.: Observer-based rejection of cogging torque disturbance for permanent magnet motors. Applied Sciences 7(9) (2017)
17. Narendra, K.S., Annaswamy, A.M.: Stable adaptive systems. Prentice-Hall (1989)
18. Niu, B., Li, H., Qin, T., Karimi, H.R.: Adaptive nn dynamic surface controller design for nonlinear pure-feedback switched systems with time-delays and quantized input. IEEE Transactions on Systems, Man, and Cybernetics: Systems 48(10), 1676–1688 (2018)
19. Rafaq, M.S., Lee, H., Park, Y., Lee, S.B., Fernandez, D., Diaz-Reigosa, D., Briz, F.: A simple method for identifying mass unbalance using vibration measurement in permanent magnet synchronous motors. IEEE Transactions on Industrial Electronics 69(6), 6441–6444 (2022)
20. Sharma, P., Ajjarapu, V., Vaidya, U.: Data-driven identification of nonlinear power system dynamics using output-only measurements. IEEE Transactions on Power Systems 37(5), 3458–3468 (2022)
21. Song, J., Wang, Y.K., Zheng, W.X., Niu, Y.: Adaptive terminal sliding mode speed regulation for pmsm under neural-network-based disturbance estimation: A dynamic-event-triggered approach. IEEE Transactions on Industrial Electronics 70(8), 8446–8456 (2023)
22. Viola, J., Chen, Y.: Parallel enabled and stability-aware self optimizing control with globalized constrained nelder-mead optimization algorithm. IEEE Journal of Radio Frequency Identification 7, 178–181 (2023)
23. Wang, H., Wang, J., Wang, X., Lu, S., Hu, C., Cao, W.: Detection and evaluation of the interturn short circuit fault in a bldc-based hub motor. IEEE Transactions on Industrial Electronics 70(3), 3055–3068 (2023)

24. Yin, K.L., Pu, Y.F., Lu, L.: Hermite functional link artificial-neural-network-assisted adaptive algorithms for iov nonlinear active noise control. IEEE Internet of Things Journal 7(9), 8372–8383 (2020)
25. Yun, Z., Quan, Z., Caixin, S., Shaolan, L., Yuming, L., Yang, S.: Rbf neural network and anfis-based short-term load forecasting approach in real-time price environment. IEEE Transactions on Power Systems 23(3), 853–858 (2008)
26. Zhai, J., Zhou, J., Zhang, L., Zhao, J., Hong, W.: Dynamic behavioral modeling of power amplifiers using anfis-based hammerstein. IEEE Microwave and Wireless Components Letters 18(10), 704–706 (2008)
27. Zhang, G., Cai, Y., Zhang, W.: Robust neural control for dynamic positioning ships with the optimum-seeking guidance. IEEE Transactions on Systems, Man, and Cybernetics: Systems 47(7), 1500–1509 (2017)
28. Zhang, J., Hou, G.: Diagonal recurrent neural networks with application to multivariable temperature control. In: 2006 1ST IEEE Conference on Industrial Electronics and Applications. pp. 1–4 (2006)
29. Zhou, J., Stamnes, O.N., Aamo, O.M., Kaasa, G.O.: Switched control for pressure regulation and kick attenuation in a managed pressure drilling system. IEEE Transactions on Control Systems Technology 19(2), 337–350 (2011)
30. Zilong, L., Guozhong, L., Jie, L.: Neural adaptive sliding mode speed tracking control of a dc motor. Journal of Systems Engineering and Electronics 15(3), 304–308 (2004)

Chao-Ting Chu graduated from the Ph.D. program in the Graduate Institute of Engineering Science and Technology at National Yunlin University of Science and Technology in 2015. Since 2016, he has been an integral part of Chunghwa Telecom Co., Ltd., specializing in IoT product service development, firmware integration, and cloud system research. Dr. Chu has been involved in a diverse range of projects, including the development of smart home systems (SmartLife), home appliances, cross-border IoT platforms, and connectivity management platforms. His contributions have significantly advanced the landscape of interconnected technologies.

Hao-Shang Ma received the B.S. and M.S. degree in Computer Science and Engineering from Yuan Ze University at 2010 and 2013 respectively. He studied in the institute of Computer and Communication Engineering for PhD in National Cheng Kung Unversity and received the PhD degree in July 2022. Currently, he is an assistant professor in Department of Computer Science and Information Engineering at National Taichung University of Science and Technology. Since January 2021, he is the Young Professionals Secretary of the Institution of Engineering and Technology (IET) - Taipei Network. His research interests include Artificial Intelligence, Data Mining, Social Network Analysis, Recommender Systems, and Nature Language Processing.

Received: August 03, 2023; Accepted: September 19, 2023.