Adaptive Recurrent Functional-Link-Based Petri Fuzzy-Neural-Network Controller for a DSP-Based Induction Motor Servo Drive System

FAYEZ F. M. EL-SOUSY*, KHALED A. ABUHASEL**
*Department of Electrical Engineering
**Department of Mechanical Engineering
College of Engineering, Salman bin Abdulaziz University
Al-KHARJ, SAUDI ARABIA
*Department of Power Electronics and Energy Conversion
*Electronics Research Institute
CAIRO, EGYPT
E-mail: *f.elsousy@sau.edu.sa, **k.abuhasel@sau.edu.sa

Abstract: In this paper, an intelligent adaptive control system (IACS) for induction motor (IM) servo drive to achieve high dynamic performance is proposed. The proposed IACS comprises a recurrent functional-link-based Petri fuzzy-neural-network (RFLPFNN) controller and a robust controller so that the developed adaptive control scheme has more robustness against parameters uncertainties and approximation errors. The RFLPFNN controller is used as the main tracking controller to mimic an ideal control law while the robust controller is proposed to compensate the difference between the ideal control law and the RFLPFNN controller. The proposed RFLPFNN model uses a functional-link neural network to the consequent part of the fuzzy rules. Thus, the consequent part of the proposed RFLPFNN model is a nonlinear combination of input variables. Moreover, the online structure and parameter-learning of the RFLPFNN are performed concurrently. The structure learning is based on the partition of input space and the parameter learning is derived based on the Lyapunov stability analysis and the back propagation method to guarantee the asymptotic stability of the IACS for the IM servo drive. In addition, to relax the requirement for the bound of minimum approximation error and Taylor higher-order terms, an adaptive control law is utilized to estimate the mentioned bounds. A computer simulation is developed and an experimental system is established to validate the effectiveness of the proposed IACS. All control algorithms are implemented in a TMS320C31 DSP-based control computer. The simulation and experimental results confirm that the IACS grants robust performance and precise response regardless of load disturbances and IM parameters uncertainties. Furthermore, the superiority of the proposed IACS is indicated in comparison with the Petri fuzzy-neural-network control system and traditional PID controller.

Key-Words: Functional-link neural-networks (FLNNs), intelligent control, indirect field-orientation control (IFOC), induction motor, Lyapunov satiability theorem, Petri net (PN), fuzzy-neural-network, robust control.

1 Introduction

Induction motors (IMs) have many advantageous characteristics such as high robustness, reliability and low cost compared with DC motors. In the last two decades, field-oriented control has become the preferred method used in the control of high performance IM drives. The objective is to obtain a torque dynamic similar to that of a separately excited DC motor. Therefore, IM drives are frequently used in high-performance industrial applications which require independent torque and speed/position control. Induction motors also possess complex nonlinear, time-varying and temperature dependency mathematical model. However, the control performance of the IM drives is sensitive to the motor parameter variations, especially the rotor time constant, which varies with the temperature and the saturation of the magnetizing inductance. In addition, the performance of IM drives is still influenced by uncertainties, such as mechanical parameter variation, external disturbance, unstructured uncertainty due to non ideal field orientation in the transient state and unmodeled dynamics. From a practical point of view, complete information about uncertainties is difficult to acquire in advance [1]-[2]. Therefore, in recent years much research has been done to apply various approaches to attenuate the effect of nonlinearities and uncertainties of IM servo drives to enhance the control performance [8]-[30]. Conventional proportional-integral-derivative (PID) controllers are widely used in industry due to their simple control structure, ease of design and implementation [3]-[7].

However, the PID controller cannot provide robust control performance because the IM servo drive system is highly nonlinear and uncertain. In addition, an objection to the real-time use of such control scheme is the lack of knowledge of uncertainties. Due to the existence of nonlinearities, uncertainties, and disturbances, conventional PID controller cannot guarantee a sufficiently high performance for the IM servo drive system. To deal with these uncertainties and nonlinearities and to enhance the control performance, many control techniques have been developed for IM drive system, such as robust control [8]-[11], sliding mode control (SMC) [12]-[16], intelligent control [17]-[24], hybrid control [25]-[28], H_∞ Control [29], [30]. These approaches improve the control performance of the IM drive from different aspects. Therefore, the motivation of this paper is to design and implement a suitable control scheme to confront the uncertainties existing in practical applications of an indirect field-oriented controlled IM drive.

The concept of incorporating fuzzy logic into a neural network (NN) has grown into a popular research topic. In contrast to the pure neural network or fuzzy system, the fuzzy-neural-network (FNN) possesses both their advantages; it combines the capability of fuzzy reasoning in handling uncertain information and the capability of NNs in learning from the process [31]-[35]. On the other hand, the recurrent fuzzy-neural-network (RFNN), which naturally involves dynamic elements in the form of feedback connections used as internal memories, has been studied in the past few years [34], [35]. In recent years, Petri net has found widely applications in modeling and controlling discrete event dynamic systems [36]-[39]. For the last decades, Petri net (PN) has developed into a powerful tool for modeling, analysis, control, optimization, and implementation of various engineering systems [40]-[46]. In [45], the concept of incorporating PN into a traditional FNN to form a new type Petri FNN (PFNN) framework for the motion control of linear induction motor drive is presented. In [46], the designed of a network structure by introducing PN into RFNN to form a dynamic Petri RFNN (DPRFNN) scheme for the path-tracking control of a nonholomonic mobile robot is presented.

One of the important points in the design of FNNs is the consequent part, which is able to impact performance on using different types. Two types of FNNs are the Mamdani-type and the Takagi-Sugeno-Kang (TSK)-type. For Mamdani-type FNNs, the minimum fuzzy implication is adopted in fuzzy reasoning. For TSK-type FNNs, the consequence part of each rule is a linear combination of input variables. It has shown that TSK-type FNN offer better network size and learning accuracy than Mamdani-type FNNs. In the TSK-type FNN, which is a linear polynomial of input variables, the model output is approximated locally by the rule hyperplanes. Nevertheless, the traditional TSK-type FNN does not take full advantage of the mapping capabilities that may be offered by the consequent part. Therefore, several researches [47]–[51] considers trigonometric functions to replace the traditional TSK-type fuzzy reasoning and also obtain better performance. In this view, the functional-link neural network (FLNN) has been proposed using trigonometric functions to construct the consequent part. The functional expansion increases the dimensionality of the input vector and thus creation of nonlinear decision boundaries in the multidimensional space and identification of complex nonlinear function become simple with this network. It seems to be more efficient to include the functional-link fuzzy rules into the PFNN. In [48]-[50], a functional-link-based fuzzy neural network for nonlinear system control is proposed, which combines a fuzzy neural network with FLNN. The consequent part of the fuzzy rules that corresponds to an FLNN comprises the functional expansion of the input variables.

With the above mention motivations, this paper presents the combination of PFNN and FLNN to construct the consequent part, called recurrent FLNN-based PFNN (RFLPFNN) controller, for dynamic system identification and control of IM servo drive system. The proposed RFLPFNN is designed to improve the accuracy of functional approximation. Each fuzzy rule that corresponds to an FLNN consists of a functional expansion of input variables. The orthogonal polynomials and linearly independent functions are adopted as FLNN bases. An online learning algorithm, consisting of structure learning and parameter learning, is proposed to construct the RFLPFNN model automatically. The structure learning algorithm determines whether or not to add a new node that satisfies the fuzzy partition of input variables. Initially, the RFLPFNN model has no rules. The rules are automatically generated from training data by entropy measure. The parameter learning algorithm is based on back propagation to tune the parameters in the RFLPFNN model simultaneously to minimize an output error function. The advantages of the proposed RFLPFNN model are summarized as follows. First, the consequent of the fuzzy rules of the proposed RFLPFNN is a nonlinear combination of input variables. This paper uses the FLNN to the consequent part of the fuzzy rules. The functional expansion in RFLPFNN can yield the consequent part of a nonlinear combination of input variables to be approximated more effectively. Second, the online learning algorithm can automatically construct the RFLPFNN. No rules or memberships exist initially. They are created automatically as learning proceeds, as online incoming training data are received and as structure and parameter learning are performed. Third, as demonstrated in Section 3, the proposed RFLPFNN can solve temporal problems effectively and is a more adaptive and efficient controller than the other methods.

This paper is organized as follows. Section 2 presents the indirect field–orientation control and dynamic analysis of the IM servo drive as well as the problem formulation. Section 3 presents the description of the intelligent adaptive control system for the IM servo drive. In addition, the design procedures and adaptive learning algorithms of the proposed RFLPFNN control system and the robust controller are described in details in Section 3. As well, the stability analysis of the proposed control system is introduced. The validity of the design procedure and the robustness of the proposed controller are verified by means of computer simulation and experimental analysis. All control algorithms have been developed in a control computer that is based on a TMS320C31 and TMS320P14 DSP DS1102 board. The dynamic performance of the IM drive system has been studied under load changes and parameter uncertainties. Numerical simulations and experimental results are provided to validate the effectiveness of the proposed control system in Section 4. Conclusions are introduced in Section 5.

2 Preliminaries

2.1 Induction Motor Dynamic Model and Indirect Field-Orientation Control

The dynamic model of the three-phase squirrel-cage IM in d-q axis arbitrary reference frame is helpful for dynamic analysis and control [1], [2]. The voltage equation in d-q axis based on the stator currents and rotor fluxes is given by (1) and the electromagnetic torque is given by (2) while the mechanical equation of the IM is given by (3) [1], [2].

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s + L_s\sigma \frac{d}{dt} & \omega\, L_s\sigma & \frac{L_m}{L_r}\frac{d}{dt} & \omega\, \frac{L_m}{L_r} \\ -\omega\, L_s\sigma & R_s + L_s\sigma \frac{d}{dt} & -\omega\, \frac{L_m}{L_r} & \frac{L_m}{L_r}\frac{d}{dt} \\ -\frac{L_m}{\tau_r} & 0 & \frac{1}{\tau_r}+\frac{d}{dt} & (\omega\, -\omega_r) \\ 0 & -\frac{L_m}{\tau_r} & -(\omega\, -\omega_r) & \frac{1}{\tau_r}+\frac{d}{dt} \end{bmatrix} \cdot \begin{bmatrix} i_{qs} \\ i_{ds} \\ \lambda_{qr} \\ \lambda_{dr} \end{bmatrix} \tag{1}$$

The electromagnetic torque can be expressed as:

$$T_e = \frac{3}{2}\frac{P_m}{2}\frac{L_m}{L_r}\left(\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}\right) \tag{2}$$

The mechanical equation can be expressed as:

$$T_e = J_m\left(\frac{2}{P_m}\right)\frac{d^2}{dt^2}\theta_r + \beta_m\left(\frac{2}{P_m}\right)\frac{d}{dt}\theta_r + T_L \tag{3}$$

The IFOC dynamics for the IM is derived from (1) and (2) respectively at the synchronous reference frame by setting $\lambda^e_{qr} = 0$ $d\lambda^e_{qr}/dt = 0$ and $\omega = \omega_e$. The torque equation and slip angular frequency for rotor flux orientation are given in (4) and (5) while the voltage commands are given in (6)-(9) [2].

$$T_e = \frac{3}{2}\frac{P_m}{2}\frac{L_m^2}{L_r} i^{e*}_{ds} i^{e*}_{qs} \tag{4}$$

$$\omega_{sl} = \frac{1}{\tau_r}\frac{i^{e*}_{qs}}{i^{e*}_{ds}} \tag{5}$$

$$V^{e*}_{qs} - e^{e*}_{qs} = \left(L_s\sigma p i^{e*}_{qs} + R_s i^{e*}_{qs}\right) \tag{6}$$

$$e^{e*}_{qs} = \left(L_s\sigma + L_m^2/L_r\right)\omega_e\, .i^{e*}_{ds} \tag{7}$$

$$V^{e*}_{ds} + e^{e*}_{ds} = \left(L_s\sigma p i^{e*}_{ds} + R_s i^{e*}_{ds}\right) \tag{8}$$

$$e^{e*}_{ds} = \left(L_s\sigma + L_m^2/L_r\right)\omega_e\, .i^{e*}_{qs} \tag{9}$$

where V_{qs}, V_{ds}, i_{qs} and i_{ds} are the d-q axis stator voltages and d-q axis stator currents, λ_{qr} and λ_{dr} are the q-axis rotor flux and d-axis rotor flux, respectively. R_s, R_r, L_s, L_r and L_m are the stator resistance, rotor resistance, stator self inductance, rotor self inductance and mutual inductance, respectively. ω_r, θ_r, J and β are the rotor speed, the rotor position, the effective inertia and the friction coefficient, respectively. T_e, T_L, τ_r, τ_s and ω_{sl} are the electromagnetic torque, the load torque, the rotor time constant, the stator time constant and the slip angular frequency, respectively. e_{qs}, e_{ds}, ω and ω_e are the back emfs, the angular frequency of the arbitrary and the synchronous reference frames respectively. σ and P_m are the leakage flux coefficient and the number of poles, respectively. $\tau_r = L_r/R_r$ and $\sigma = (L_s L_r - L_m^2)/L_s L_r$. The parameters of the IM are listed in Table (1).

The dynamic analysis of the IM may be expressed as follows. From (3) and (4), the mechanical dynamics can be simplified as:

$$\ddot{\theta}_r = -\frac{\beta_m}{J_m}\frac{2}{P_m}\dot{\theta}_r + \frac{K_t}{J_m}\frac{2}{P_m} i^{e*}_{qs} - \frac{2}{P_m}\frac{1}{J_m}T_L \tag{10}$$

$$\ddot{\theta}_r = A_m\dot{\theta}_r + B_m U(t) + D_m T_L \tag{11}$$

where $U(t) = i^{e*}_{qs}(t)$ is the control effort, $K_t = (3/2)(P_m/2)(L_m^2/L_r)i^{e*}_{ds}$, $A_m = -(\beta_m/J_m)(P_m/2)$, $B_m = K_t/(J_m)(P_m/2)$ and $D_m = -(P_m/2)(1/J_m)$.

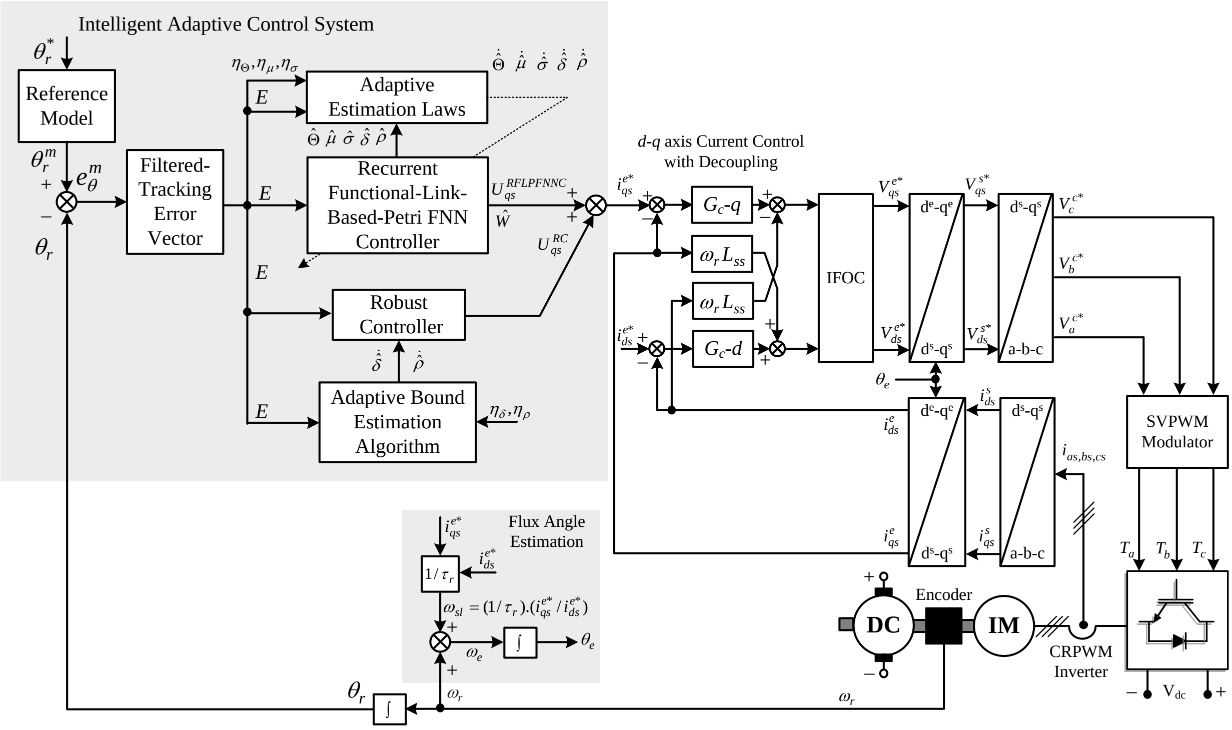


Fig. 1 Structure of the proposed intelligent adaptive control system (IACS) with RFLPFNN for IM servo drive

2.2 Problem Formulation

In order to efficiently control the rotor position of the IM drive system, an IACS is proposed to increase the robustness of the IFOC-IM drive system. The intelligent adaptive control system for the IM drive is shown in Fig. 1, in which the reference model is used as the closed loop transfer function of the drive system with the 2DOF I-PD position controller [20]. The proposed controller combines a RFLPFNN controller and a robust controller. The RFLPFNN controller is utilized as the main tracking controller to mimic an ideal control law while the robust controller is designed with adaptive bound estimation algorithm to recover the residual of the approximation error via the RFLPFNN controller approximation. The hybrid control law is assumed to take the following form:

$$U_{qs}(t)=U_{qs}^{RFLPFNNC}(t)+U_{qs}^{RC}(t) \qquad (12)$$

where $U_{qs}^{RFLPFNNC}(t)$ is the RFLPFNN controller and $U_{qs}^{RC}(t)$ is the robust controller. In the following section, the description of the proposed control scheme and the stability analysis are introduced.

3 Intelligent Adaptive Control System (IACS)

In this section, the design procedure for the recurrent functional-link-based Petri fuzzy-neural-network controller in order to control the position of the IM is carried out. The proposed RFLPRFNN control scheme combines the merits of the Petri fuzzy-neural-network (PFNN), recurrent FNN and FLNN so that the robust control performance of the IM servo drive can be preserved. Define the tracking error vector as follows:

$$E=[(\theta_r^m-\theta_r)\quad(\dot\theta_r^m-\dot\theta_r)]^T=[e_\theta^m\quad \dot e_\theta^m]^T \qquad (13)$$

where $\theta_r^m(t)$ is the desired position command, $\theta_r(t)$ is the actual rotor position, $e_\theta^m(t)=[\theta_r^m(t)-\theta_r(t)]$ is the position error and $\dot e_\theta^m(t)=[\dot\theta_r^m(t)-\dot\theta_r(t)]$ is the tracking position error change. Now, assume that the parameters of the IM servo drive system and the external load disturbance are well known, the ideal control law can be defined as follows:

$$U_{qs}^*(t)=B_m^{-1}[\ddot\theta_r^m(t)-A_m\dot\theta_r(t)-D_mT_L(t)+KE] \qquad (14)$$

where $K=[k_2\quad k_1]$, in which k_1 and k_2 are positive constants. Substituting (14) into (11) will yield

$$\ddot e_\theta^m(t)+k_1\dot e_\theta^m(t)+k_2 e_\theta^m(t)=0 \qquad (15)$$

Suppose the control gain K is chosen such that all roots of the characteristic polynomial of (15) lie strictly in the open left half of the complex plane. This implies that the position tracking error will converge to zero when time tends to infinity, i.e. the IM servo drive states can track the desired trajectory asymptotically. However, the parameter variations of the IM are difficult to measure and the exact value of the external load disturbance is also difficult to know in advance for practical applications. Though, if the IM parameters are perturbed, the performance specified by (15) can not guarantee. Moreover, the stability of the IM servo drive may be destroyed. Therefore, to ensure the stability of the IM servo drive despite the existence of the uncertain dynamics and external load disturbance, a RFLPFNN controller is proposed.

Table (1) Parameters of IM servo drive used in simulation and experimentation

Quantity	Symbol	Value
Nominal power	P_n	1.5 kW (3-phase)
Rated voltage	V_{L-L}	380 V
Nominal speed (electrical)	ω_r	296.48 rad/sec
Nominal speed (mechanical)	N_r	1415 rpm
Number of poles	P_m	4
Rated torque	T_e	12 N.m
Rated current	I	3.8 A
Rated frequency	f	50 Hz
Self inductance	$L_s = L_r$	480 mH
Magnetizing inductance	L_m	464 mH
Stator winding resistance	R_s	6.30 Ω
Rotor resistance	R_r	3.60 Ω
Rotor inertia	J_m	0.038 kg.m^2
Friction coefficient	β_m	0.0085 N.m/rad/sec
Resolution of the encoder	n_E	5000 p/r

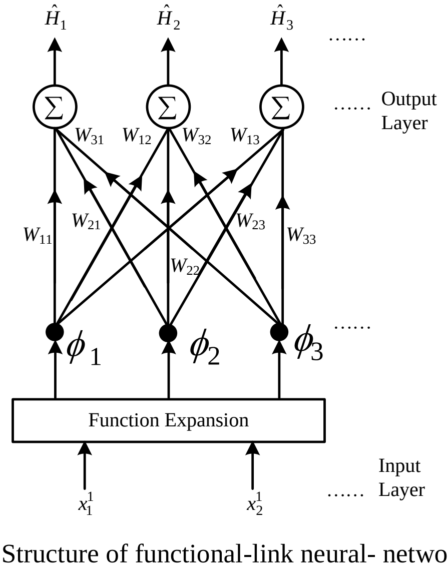


Fig. 2 Structure of functional-link neural- network

The following subsections describe the structure of FLNNs and RFLPFNN models. In FLNNs, the input data usually incorporate high-order effects, and thus, artificially increase the dimensions of the input space using a functional expansion. Consequently, the input representation is enhanced and linear reparability is achieved in the extended space. The RFLPFNN adopted the FLNN and generating complex nonlinear combinations of input variables to the consequent part of the fuzzy rules. The details of these structures are given below.

3.1 Structure of Functional-Link Neural-Network (FLNN)

To improve the accuracy of the function approximation, an FLNN is adopted to implement the function expansion for the proposed RFLPFNN. The architecture of the FLNN is shown in Fig. 2, in which the functional link acts on an element of input variables by generating a set of linearly independent functions, i.e., function expansion. In the function expansion of the FLNN, basis functions such as trigonometric, Gaussian, or other orthogonal polynomials are feasible. In this paper, the trigonometric function is adopted since it forms a more compact representation than the Gaussian basis, and the sine and cosine functions can be computed more quickly. Moreover, better performance results when the outer product term is taken into account in the function expansion [47]. Therefore, for a two-dimensional input vector $\mathrm{X}=[x_1, x_2]^T$, the input variables are obtained using trigonometric functions and can be separated in the enhanced space $\Phi=[\phi_1,\phi_2,\ldots,\phi_M]^T =$ $[1, x_1, \sin(\pi x_1), \cos(\pi x_1), x_2, \sin(\pi x_2),$ $\sin(\pi x_2), \cos(\pi x_2), x_1 x_2]^T$, where $x_1 x_2$ is the outer product term. In the FLNN structure shown in fig. 2, a set of basis functions Φ and a fixed numbers weight parameters W represent $f_W(x)$. The theory behind the FLNN for multidimensional function approximation has been discussed in [47], [51]. Let $B=\{\phi_k\}_{k=1}^{M}$ be a set of basis functions to be considered as shown in Fig. 2. The FLNN comprises M basis functions $\{\phi_1,\phi_2,\ldots,\phi_M\}\in B_M$. The output of the FLNN is given by a linear sum of the jth node as

$$\begin{aligned}\hat{H}_j &= \sum_M \omega_{Mj}\Phi_M(\mathrm{X}) \\ &= \omega_{1j}\phi_1+\omega_{2j}\phi_2+\omega_{3j}\phi_3+\cdots+\omega_{Mj}\phi_M \qquad (16)\\ &, \ M=1,2\cdots,8 \ \text{ and } \ j=1,\cdots,m\end{aligned}$$

where $\mathrm{X}\in\Re^n$, $\mathrm{X}=[x_1,x_2,\ldots,x_n]^T$ is the input vector and $W_j=[\omega_{1j},\omega_{2j},\ldots,\omega_{Mj}]^T$ is the weight vector associated with the jth output of the FLNN. $\hat{H}_j$ denotes the local output of the FLNN structure and the consequent part of the jth fuzzy rule in the RFLPFNN. Therefore, the matrix form of (16) can be expressed as

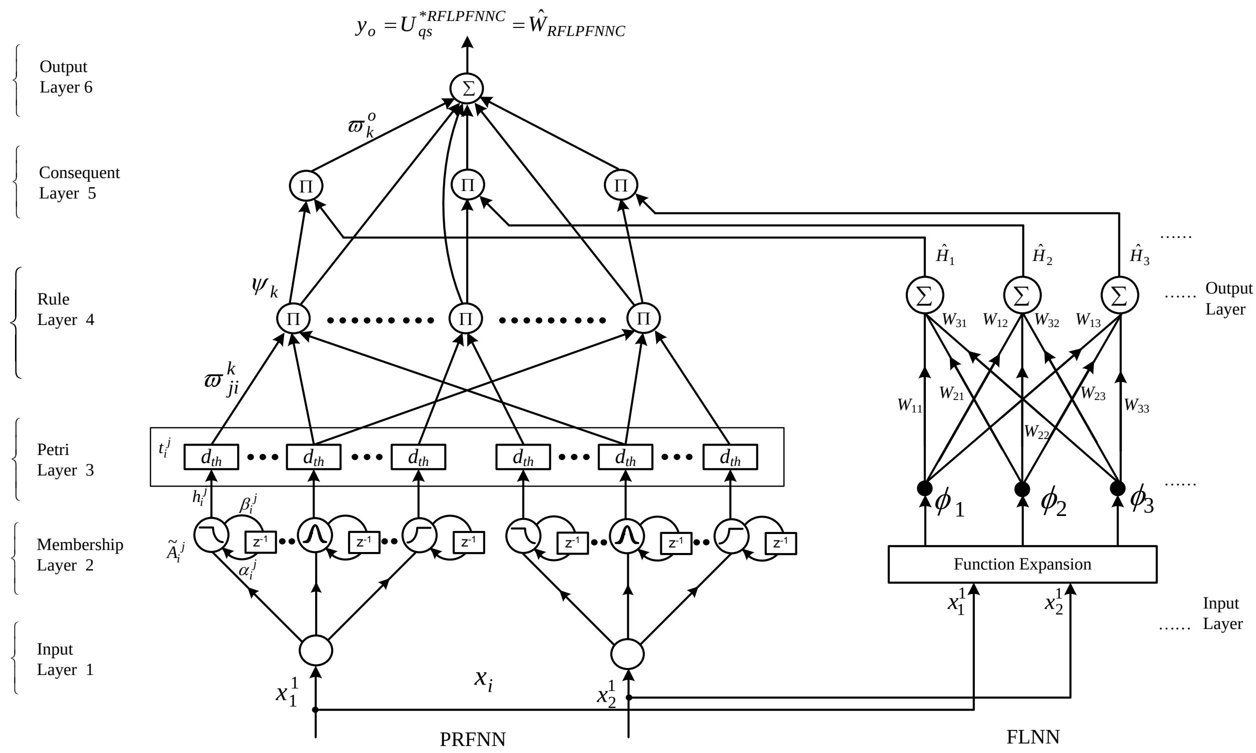


Fig. 3 Structure of recurrent functional-link-based Petri fuzzy-neural-network (RFLPFNN)

$$\hat{H}_j = W_j \Phi \tag{17}$$

where $\Phi = [\phi_1(x), \phi_2(x), \ldots, \phi_n(x)]^T$ is the basis function vector which is the output of the functional expansion block. In the RFLPFNN, the corresponding weights of functional link bases do not exist in the initial state, and the amount of the corresponding weights of functional link bases generated by the online learning algorithm is consistent with the number of fuzzy rules. The details of the online learning algorithm will be given in subsection 3.3.

3.2 Structure of Recurrent Functional-Link-Based Petri Fuzzy-Neural-Network (RFLPFNN)

This subsection describes the RFLPFNN, which uses a nonlinear combination of input variables (FLNN). Each fuzzy rule corresponds to a sub-FLNN, comprising a functional link. Fig. 3 represents the structure of the proposed RFLPFNN.

The RFLPFNN model realizes a fuzzy IF-THEN rule in the following form:

Rule *j*:
IF x_1^1 is $\tilde{A}_1^j$ and x_2^1 is $\tilde{A}_2^j$ ... and x_i^j is $\tilde{A}_i^j$ ... and x_n^j is $\tilde{A}_n^j$ THEN $\hat{H}_j = \sum_M \omega_{Mj} \Phi_M(\mathrm{X})$

$$= \omega_{1j}\phi_1 + \omega_{2j}\phi_2 + \omega_{3j}\phi_3 + \cdots + \omega_{Mj}\phi_M$$

$$y_o = \sum_{k=1}^{N_r} \varpi_k^o . \psi_k . \sum_M \omega_{Mj} \Phi_M \tag{18}$$

where x_i^j and $\hat{H}_j$ are the input and local output variables, $\tilde{A}_i^j$ is the linguistic term of the precondition part with a Gaussian membership function, *n* is the number of input variables, ω_{Mj} is the connective weight of the local output, Φ_M is the basis function of the input variables, *M* is the number of basis function, and the Rule *j* is the *j*th fuzzy rule. The connective weight ϖ_k^o is the output action strength of the *o*th output associated with the *k*th rule and ψ_k represents the output of the rule layer. The operation functions of the nodes in each layer of the RFLPFNN model are described.

The architecture of the proposed six-layer RFLPRFNN is shown in Fig. 3, which comprises the input layer (the *i* layer), membership layer (the *j* layer), Petri layer (the *p* layer), rule layer (the *k* layer), consequent layer (the *l* layer) and output layer (the *o* layer). The signal propagation and the basic function in each layer are introduced as follows.

1) Layer 1- Input Layer: The nodes in layer 1 transmit the input signals to the next layer. For every node i in the input layer, the net input and the net output can be represented as:

$$net_i^1(n) = x_i^1, \quad y_i^1 = f_i^1(net_i^1(n)) = net_i^1(n) \quad i = 1, 2 \tag{19}$$

where $x_1^1 = e_\theta^m(t)$, $x_2^1 = \dot{e}_\theta^m(t)$ and x_i^1 represents the i*th* input to the node of layer 1, *n* denotes the number of iterations.

2) Layer 2- Membership Layer: Each node in this layer performs a membership function. The input of the membership layer can be represented by

$$h_i^j(n) = x_i(n) + \beta_i^j(n-1)\alpha_i^j \tag{20}$$

where α_i^j represents the weight of the self-feedback loop, $\beta_i^j(n-1)$ indicates the output signal of layer 2 in the previous time and is defined with Gaussian membership function as

$$net_j(h_i^j) = -\left(\frac{h_i^j - \mu_i^j}{\sigma_i^j}\right)^2 \tag{21}$$

$$\beta_i^j\left[net_j(h_i^j)\right] = \exp\left[net_j(h_i^j)\right] = \exp\left[-\left(\frac{h_i^j - \mu_i^j}{\sigma_i^j}\right)^2\right] \tag{22}$$

where exp[·] is the exponential function, μ_i^j and σ_i^j (i = 1,…, n_i; j = 1,…, n_j), respectively, are the mean and standard deviation of the Gaussian function in the jth term of the ith input variable x_i to the node of this layer, n_j is the number of linguistic variables with respect to each input.

3) Layer 3- Petri Layer: The Petri layer of the RFLPRFNN is used to produce tokens and also makes use of competition learning laws to select suitable fired nodes as follows

$$t_i^j = \begin{cases} 1, & \beta_i^j(net_j(h_i^j)) \geq d_{th} \\ 0, & \beta_i^j(net_j(h_i^j)) < d_{th} \end{cases} \tag{23}$$

where t_i^j is the transition and d_{th} is a dynamic threshold value varied with the corresponding error to introduced later.

4) Layer 4- Rule Layer: Each node k in layer 4 (rule layer) is denoted by ∏, which multiplies the incoming signals and outputs the result of the product. For the kth nodes:

$$net_k = \begin{cases} \prod_{i=1}^{n} \varpi_{ji}^k \beta_i^j(net_j(h_i^j)), & t_i^j = 1 \\ 0, & t_i^j = 0 \end{cases} \tag{24}$$

From (22) and (24), we can obtain

$$\psi_k = f_k(net_k) = \begin{cases} \prod_{i=1}^{n} \varpi_{ji}^k \exp\left[-\left(\frac{h_i^j - \mu_i^j}{\sigma_i^j}\right)^2\right], & t_i^j = 1 \\ 0, & t_i^j = 0 \end{cases} \tag{25}$$

$$k = 1, \ldots, N_r$$

where β_i^j represents the jth input to the node of the rule layer (layer 4), ψ_k represents the kth output of the rule layer; ϖ_{ji}^k, the connective weights between the Petri layer and the rule layer. These weights are also assumed to be unity; and N_r is the total number of rules.

5) Layer 5-Consequent Layer: Nodes in this layer are called consequent nodes. The input to a node in layer 5 is the output from layer 4 and the other inputs are nonlinear combinations of input variables from the FLNN $\hat{H}_j$ as shown in Fig. 3. For such node

$$y_j^l = \psi_k . \hat{H}_j = \psi_k . \sum_M \omega_{Mj} \Phi_M$$
$$= \begin{cases} \sum_M \omega_{Mj} \Phi_M . \prod_{i=1}^{n} \varpi_{ji}^k \beta_i^j(net_j(h_i^j)), & t_i^j = 1 \\ 0, & t_i^j = 0 \end{cases} \tag{26}$$

where ω_{Mj} is the corresponding connective weight of the FLNN and Φ_M is the functional expansion of input variables. The functional expansion uses a trigonometric polynomial basis function, given by $[x_1 \sin(\pi x_1) \cos(\pi x_1) x_2 \sin(\pi x_2) \cos(\pi x_2)]$ for two-dimensional input variables. Moreover, the output nodes of the FLNN depend on the number of fuzzy rules of the RFLPFNN model.

6) Layer 6- Output Layer: Each node in this layer corresponds to a single output variable. The output node in layer 6 integrates all of the actions from layer 4 and 5 and acts as a center of area defuzzifier.

$$net_o = \sum_{k=1}^{N_r} \varpi_k^o . \psi_k . \sum_M \omega_{Mj} \Phi_M \tag{27}$$

$$y_o = f_o(net_o) = net_o \quad o = 1 \tag{28}$$

$$y_o = \frac{\hat{H}_j . \sum_{k=1}^{N_r} \varpi_k^o . \psi_k}{\sum_{k=1}^{N_r} \varpi_k^o . \psi_k} = \frac{\sum_{k=1}^{N_r} \varpi_k^o . \psi_k . \sum_M \omega_{Mj} \Phi_M}{\sum_{k=1}^{N_r} \varpi_k^o . \psi_k} \tag{29}$$

where the connective weight ϖ_k^o is the output action strength of the oth output and y_o is the output of the RFLPFNN and N_r is the total number of rules.

The output of the RFLPFNN controller, $y_o = U_{qs}^{RFLPFNNC}(t)$, can be rewritten as

$$U_{qs}^{RFLPFNNC} = W_{RFLPFNNC}(E, \Theta, \mu, \sigma) = \Theta^T \Omega \tag{30}$$

where the tracking error vector E is the input of the RFLPFNN, Θ are the collections of the adjustable parameters $(\omega_{Mj}, \alpha_i^j, \varpi_k^o)$, Ω is the output vector (ψ_k, Φ_M) and $E = [e_\theta^m \quad \dot{e}_\theta^m]^T$ is the tracking error input vector to the RFLPFNN controller.

3.3 Learning Algorithms of the Recurrent FLNN-based PFNN Model

This section presents an online learning algorithm for constructing the recurrent functional-link-based Petri fuzzy-neural-network model. The proposed learning algorithm comprises structure learning and a parameter learning algorithms. Structure learning is based on the entropy measure used to determine whether a new rule should be added to satisfy the fuzzy partitioning of input variables. Parameter learning is based on supervised learning algorithms. The backpropagation algorithm minimizes a given cost function by adjusting the link weights in the consequent part and the parameters of the membership functions. Initially, there are no nodes in the network except the input–output nodes, i.e., there are no nodes in the RFLPFNN model. The nodes are created automatically as learning proceeds, upon the reception of online incoming training data in the structure and parameter learning processes.

3.3.1 Structure Learning Algorithm

The structure learning algorithm is responsible for on-line rule generation. The first step in structure learning is to determine when to generate a new rule. The way the input space is partitioned determines the number of rules extracted from the training data, as well as the number of fuzzy sets in the universe of discourse for each input variable because one cluster in the input space corresponds to one fuzzy rule, in which μ_i^j represents the mean and σ_i^j represents the variance of that cluster, respectively. For each incoming pattern x_i, the rule firing strength can be regarded as the degree to which the incoming pattern belongs to the corresponding cluster [52]. For computational efficiency, the degree measure can be calculated using the firing strength from (25) as

$$D_j = \psi_j, \quad j = 1, \ldots, N(t) \tag{31}$$

where $N(t)$ denotes the number of existing rules at time t and $D_j \in [0,1]$. According to the degree measure, the criterion for the generation of a new fuzzy rule and a new FLNN base for a new incoming data is described as follows. Find the maximum degree $D_{\max}$

$$D_{\max} = \max_{1 \le j \le N(t)} D_j \tag{32}$$

If $D_{\max} \le D$, then a new rule and a new FLNN base are generated, where $D \in [0,1]$ is a prescribed threshold that decays during the learning process, thus limiting the size of the RFLPFNN. Once a new rule has been generated, the next step is to assign the initial mean and variance to the new membership function and the corresponding connective weight for the consequent part. Since the goal is to minimize an objective function, the mean, variance and weight are all adjustable later in the parameter learning phase. Hence, the mean, variance, and weight for the new rule are set as follows

$$\mu_{ij}^{N(t+1)} = x_i(t) \tag{33}$$

$$\sigma_{ij}^{N(t+1)} = \sigma_{init} \tag{34}$$

$$\omega_{Mj}^{N(t+1)} = r \in [-1,+1] \tag{35}$$

$$\alpha_{ij}^{N(t+1)} = r \in [-1,+1] \tag{36}$$

$$\varpi_{ko}^{N(t+1)} = r \in [-1,+1] \tag{37}$$

where x_i is the new input and σ_{init} is a prespecified constant. The whole algorithm for the generation of new fuzzy rules and fuzzy sets in each input variable is as follows. No rule is assumed to exist initially.

```
Step 1: IF $x_i$ is the first incoming pattern THEN do
        {Generate a new rule
        with mean $\mu_{i1} = x_i$, variance $\sigma_{i1} = \sigma_{init}$,
        weight of feedback $\alpha_{i1} = r \in [-1,+1]$,
        weight of consequent part $\omega_{M1} = r \in [-1,+1]$,
        weight $\varpi_{ko} = r \in [-1,+1]$
        where $\sigma_{init}$ is a prespecified constant.
        }
Step 2: ELSE for each newly incoming $x_i$, do
        {Find $D_{\max} = \max_{1 \le j \le N(t)} D_j$
        IF $D_{\max} \ge D$
        do nothing
        ELSE
        $N(t+1)=N(t)+1$
        generate a new rule
        with mean $\mu_{ij}^{N(t+1)} = x_i$, variance
        $\sigma_{ij}^{N(t+1)} = \sigma_{init}$,
        weight of feedback $\alpha_{ij}^{N(t+1)} = r \in [-1,+1]$,
        weight of consequent part
        $\omega_{Mj}^{N(t+1)} = r \in [-1,+1]$,
        weight $\varpi_{ko}^{N(t+1)} = r \in [-1,+1]$
        where $\sigma_{init}$ is a prespecified constant.}
        }
```

3.3.2 Parameter Learning Algorithm

The parameter learning is based on supervised learning algorithm to adjust the connected weights in the consequent part, the feedback weights and the parameters of the membership functions using the backpropagation algorithm to minimize a given energy function. To describe the on-line parameter learning algorithm of the recurrent functional-link-based Petri fuzzy-neural-network, first the energy function E_θ is defined as

$$E_\theta = \frac{1}{2}[\theta_r^m - \theta_r]^2 = \frac{1}{2}[e_\theta^m]^2 \tag{38}$$

where $\theta_r^m(t)$ is the desired position command, $\theta_r(t)$ is the actual rotor position. In each training cycle, starting at the input variables, a forward pass is adopted to calculate the activity of the model output

$y_o(t)$. The dynamic threshold value in (23) is tuned by the following relation:

$$d_{th} = \frac{\gamma \exp(-\xi E_\theta)}{1+\exp(-\xi E_\theta)} \tag{39}$$

where γ and ξ are positive constants. If the tracking errors become large, the threshold values will be decreased in order to fire more control rules for the present circumstances [46].

When the backpropagation learning algorithm is adopted, the weighting vector of the RFLPFNN model is adjusted such that the error defined in (38) is less than the desired threshold value after a given number of training cycles. The learning algorithm based on the well-known backpropagation is described as follows.

1) Layer 6: In the output layer (layer 5), the error term to be propagated is calculated as:

$$\delta_o^{(6)} = -\frac{\partial E_\theta}{\partial y_o^{(6)}} = -\left[\frac{\partial E_\theta}{\partial e_\theta^m}.\frac{\partial e_\theta^m}{\partial y_o^{(6)}}\right] = -\left[\frac{\partial E_\theta}{\partial e_\theta^m}.\frac{\partial e_\theta^m}{\partial \theta_r}.\frac{\partial \theta_r}{\partial y_o^{(6)}}\right]$$
$$= \left(e_\theta^m \frac{\partial \theta_r}{\partial y_o^{(6)}}\right) \tag{40}$$

The weight is updated by the amount:

$$\Delta \varpi_k^o = -\eta_\varpi \frac{\partial E_\theta}{\partial \varpi_k^o}$$
$$= \left[-\eta_\varpi \frac{\partial E_\theta}{\partial y_o^{(6)}}\right]\left(\frac{\partial y_o^{(6)}}{\partial net_o^{(6)}}.\frac{\partial net_o^{(6)}}{\partial \varpi_k^o}\right) = \eta_\varpi \delta_o^{(6)} \psi_k \tag{41}$$

where η_ϖ is the learning rate parameter of the connecting weights of the output layer.

The weights of the output layer (layer 6) are updated according to the following equation.

$$\varpi_k^o(t+1) = \varpi_k^o(t) + \Delta \varpi_k^o \tag{42}$$

where t denotes the iteration number of the jth.

2) Layer 5: In the consequent layer (layer 5), the error term to be propagated is calculated as:

$$\delta_j^{(5)} = -\frac{\partial E_\theta}{\partial y_j^l} = -\left[\frac{\partial E_\theta}{\partial y_o^{(6)}}.\frac{\partial y_o^{(6)}}{\partial y_j^l}\right] = \delta_o^{(6)} \frac{1}{\sum_{k=1}^{N_r} \varpi_k^o.\psi_k} \tag{43}$$

According to (43), the weights in the FLNN are updated using the following equation.

$$\Delta \omega_{Mj} = -\eta_\omega \frac{\partial E_\theta}{\partial \omega_{Mj}} = \left[-\eta_\omega \frac{\partial E_\theta}{\partial y_j^l}\right]\left[\frac{\partial y_j^l}{\partial \omega_{Mj}}\right] \tag{44}$$
$$= \eta_\omega \delta_j^{(5)}.y_j^{(4)}.\Phi_M = \eta_\omega \delta_j^{(5)}.\psi_k.\Phi_M$$

where η_ω is the learning rate parameter of the connecting weights of the FLNN.
The weights of the consequent layer (layer 5) are updated according to the following equation.

$$\omega_{Mj}(t+1) = \omega_{Mj}(t) + \Delta \omega_{Mj} \tag{45}$$

3) Layer 4: In rule layer (layer 4), only the error term needs to be computed and propagated because the weights in this layer are unity.

$$\delta_k^{(4)} = -\frac{\partial E_\theta}{\partial net_k^4} = -\frac{\partial E_\theta}{\partial \psi_k^4} = \left[-\frac{\partial E_\theta}{\partial y_j^l}\frac{\partial y_j^l}{\partial \psi_k^4}\right]$$
$$= \begin{cases} \delta_j^{(5)} \sum_M \omega_{Mj} \Phi_M, & \psi_k^4 \neq 0 \\ 0, & \psi_k^4 = 0 \end{cases} \tag{46}$$

4) Layer 3: In the Petri layer (layer 3), the error term is calculated as follows:

$$\delta_j^{(3)}(h_i^j) = -\frac{\partial E_\theta}{\partial net_j^3(h_i^j)} = \left(-\frac{\partial E_\theta}{\partial y_j^l}\frac{\partial y_j^l}{\partial \psi_k^4}\right)\left(\frac{\partial \psi_k^4}{\partial \beta_i^j(net_j^3(h_i^j))}\right).$$
$$\left(\frac{\partial \beta_i^j(net_j^3(h_i^j))}{\partial net_j^3(h_i^j)}\right) = \begin{cases} \sum_k \delta_k^{(4)} \psi_k^4 & t_i^j = 1 \\ 0, & t_i^j = 0 \end{cases} \tag{47}$$

The update laws of μ_i^j, σ_i^j and α_i^j are given by:

$$\Delta \mu_i^j = -\eta_\mu \frac{\partial E_\theta}{\partial \mu_i^j} = \eta_\mu \left[-\frac{\partial E_\theta}{\partial y_j^l}\frac{\partial y_j^l}{\partial net_j^3(h_i^j)}\frac{\partial net_j^3(h_i^j)}{\partial \mu_i^j}\right]$$
$$= \eta_\mu \delta_j^{(3)}(h_i^j)\frac{2(h_i^j - \mu_i^j)}{(\sigma_i^j)^2} \tag{48}$$

$$\Delta \sigma_i^j = -\eta_\sigma \frac{\partial E_\theta}{\partial \sigma_i^j} = \eta_\sigma \left[-\frac{\partial E_\theta}{\partial y_j^l}\frac{\partial y_j^l}{\partial net_j^3(h_i^j)}\frac{\partial net_j^3(h_i^j)}{\partial \sigma_i^j}\right]$$
$$= \eta_\sigma \delta_j^{(3)}(h_i^j)\frac{2(h_i^j - \mu_i^j)^2}{(\sigma_i^j)^3} \tag{49}$$

$$\Delta \alpha_i^j = -\eta_\alpha \frac{\partial E_\theta}{\partial \alpha_i^j} = \eta_\sigma \left[-\frac{\partial E_\theta}{\partial y_j^l}\frac{\partial y_j^l}{\partial net_j^3(h_i^j)}\frac{\partial net_j^3(h_i^j)}{\partial \alpha_i^j}\right]$$
$$= \eta_\alpha \delta_j^{(3)}(h_i^j)\frac{2(h_i^j - \mu_i^j)}{(\sigma_i^j)^2} \tag{50}$$

where $\eta_\mu,$ η_σ and η_α are the learning rate parameters of the mean and variance and the self-feedback loop, respectively. Moreover, they can updated as follows:

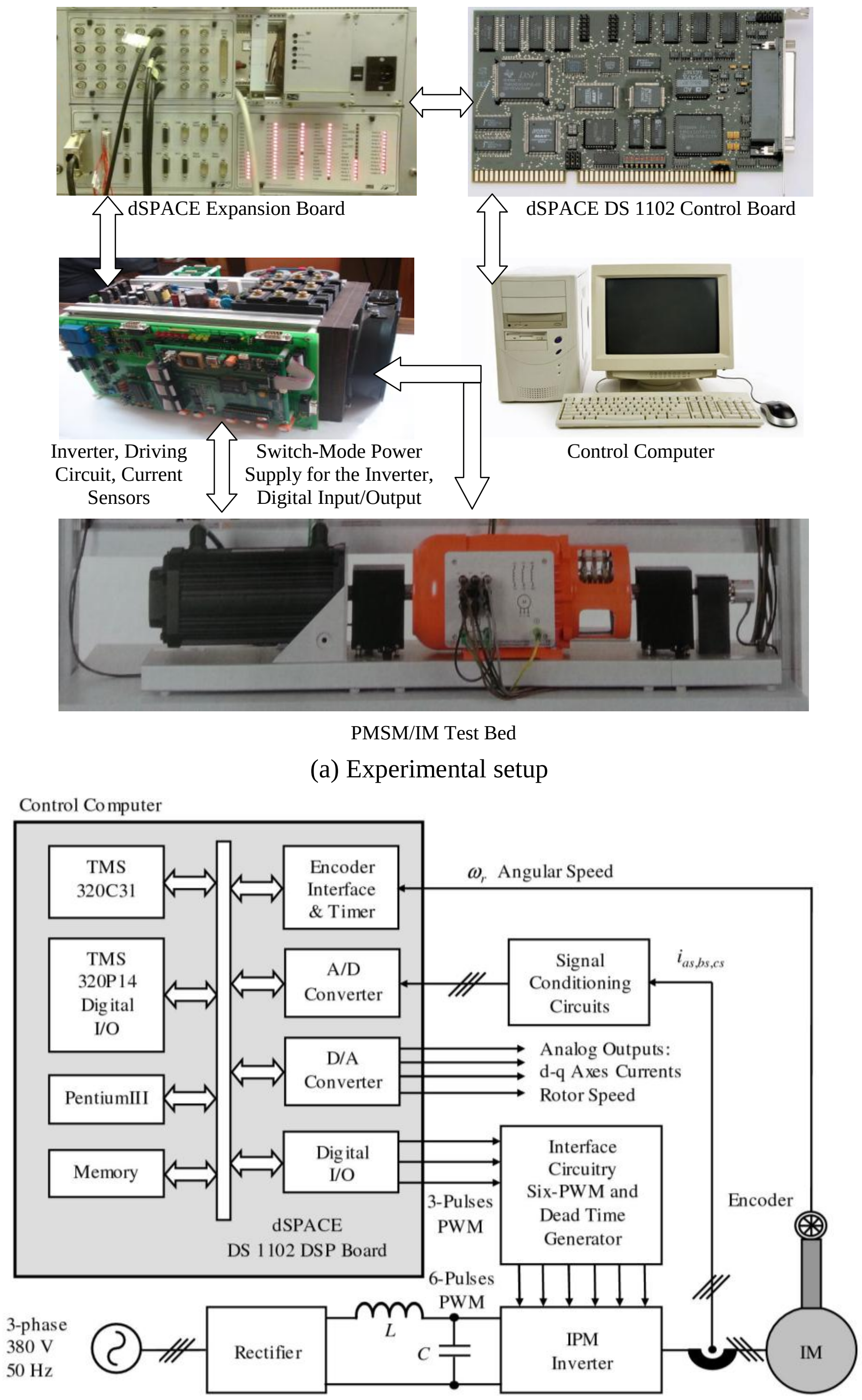
$$\mu_j^i(t+1) = \mu_j^i(t) + \Delta \mu_j^i(t) \tag{51}$$

$$\sigma_j^i(t+1) = \sigma_j^i(t) + \Delta \sigma_j^i(t) \tag{52}$$

$$\alpha_j^i(t+1) = \alpha_j^i(t) + \Delta \alpha_j^i(t) \tag{53}$$

The exact calculation of the IM servo drive system Jacobian, $\partial \theta_r / \partial y_o^{(6)} = \partial \theta_r / \partial i_{qs}^{e*}$, in (40) can not be determined due to the uncertainties of the IM dynamics. To overcome this problem and to increase the online learning rate of the network parameters, the delta adaptation law is adopted as follows:

$$\delta_o^{(6)} = e_\theta^m + k_\theta^m \dot{e}_\theta^m \tag{54}$$



(a) Experimental setup

(b) Block diagram of the proposed DSP-based control system

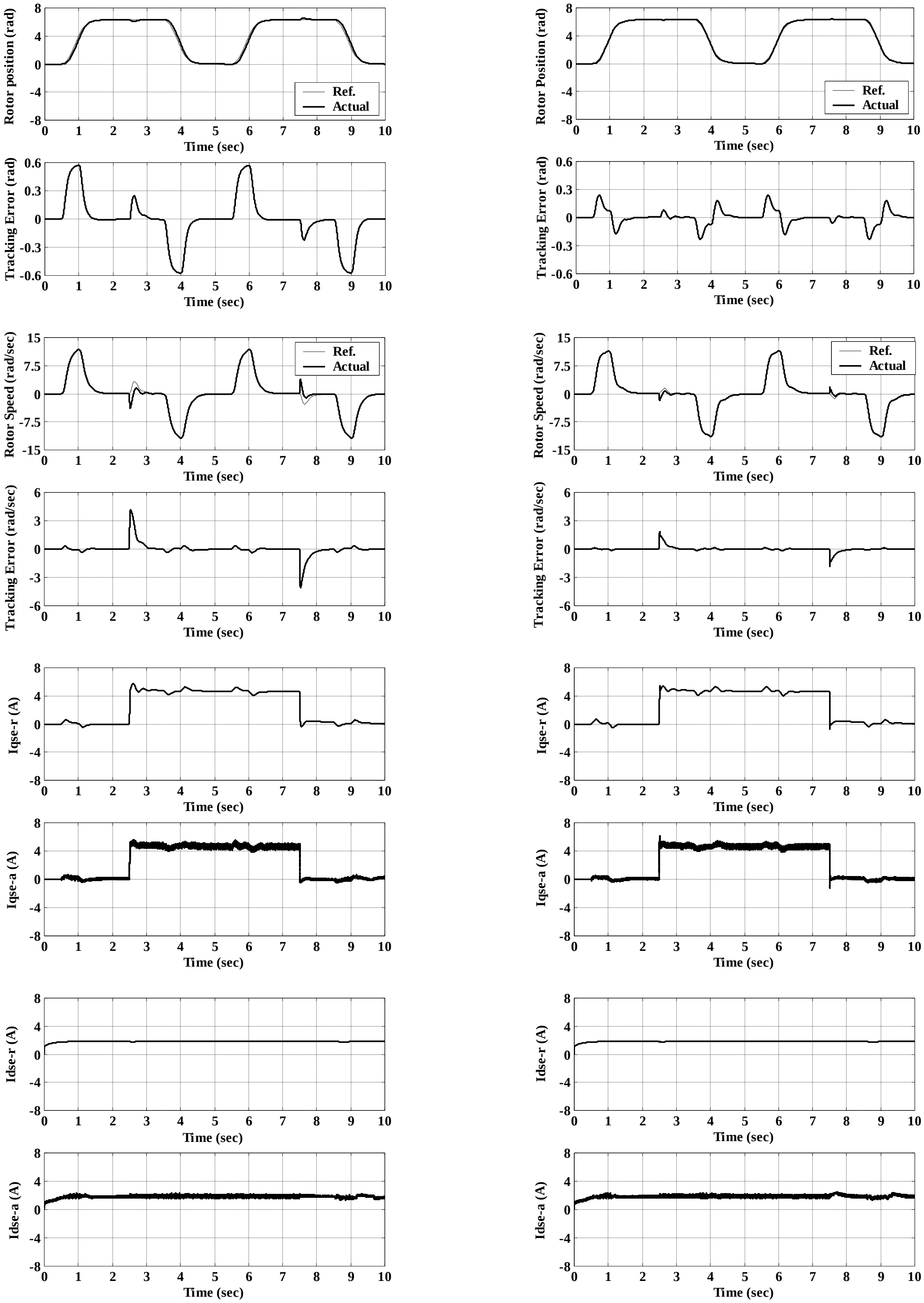
Fig. 4 DSP-based intelligent adaptive control system (IACS) using RFLPFNN for IM servo drive

3.4 Robust Control System Design and Stability Analysis of the IACS

In this section, the intelligent adaptive control system design and stability analysis is carried out. The proposed controller combines the merits of the PFNN, recurrent neural network and FLNN. The design of robust controller is necessary to compensate the minimum approximation error instead of increasing the rules of the RFLPFNN controller. From (11), (12) and (14), an error equation is obtained as follows:

$$\dot{E} = \Lambda E + \bar{B}_m\left(U_{qs}^{*} - \hat{U}_{qs}^{RFLPFNNC} - U_{qs}^{RC}\right) \qquad (55)$$

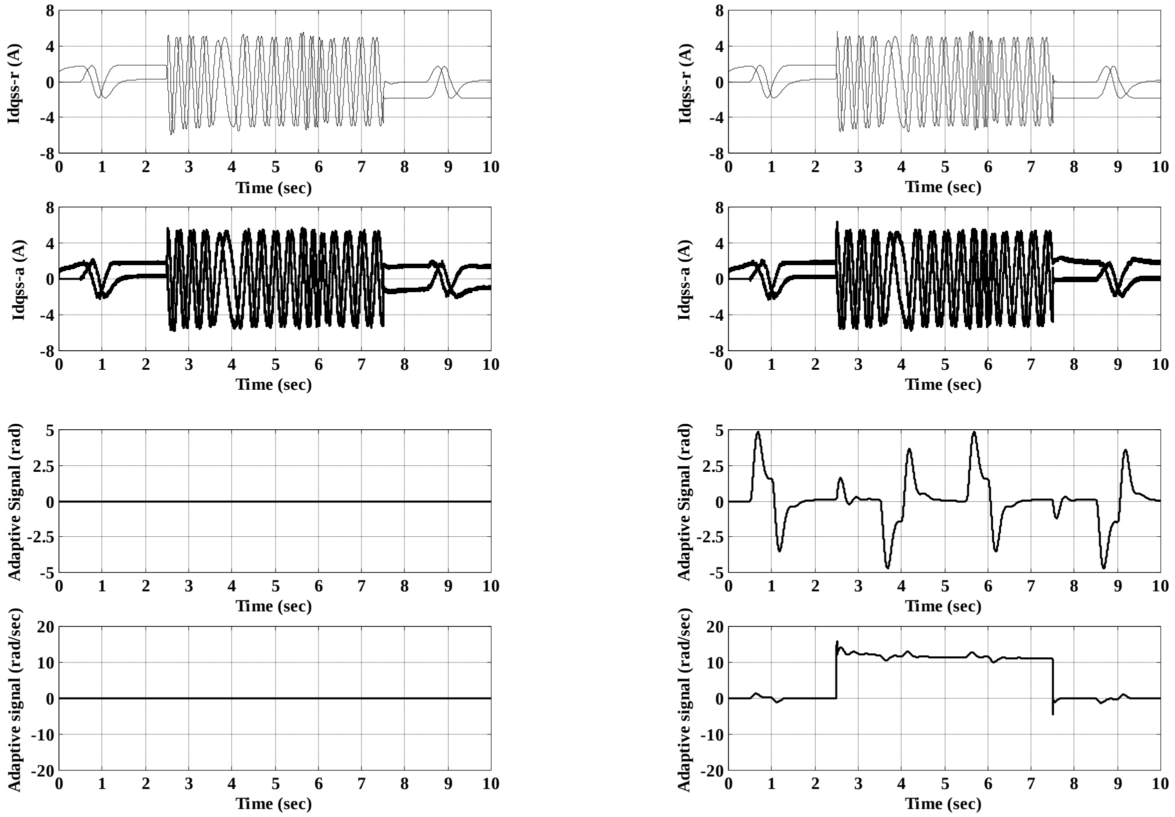
where $\Lambda = \begin{bmatrix} 0 & 1 \\ -k_2 & -k_1 \end{bmatrix}$ is a stable matrix and $\bar{B}_m = [0 \quad B_m]^T$. To develop the robust controller, a minimum approximation error ε is defined as follows:



(a) (b)

Fig. 5 Dynamic response of the IM servo drive system for the reference position and subsequent loading of 12 N.m for both position controllers at Case (1) of parameter uncertainties.

(a) Using PFNN position tracking controller (b) Using IACS with RFLPFNN position tracking controller



(a) (b)

Fig. 5 (*Continued*) Dynamic response of the IM servo drive system for the reference position and subsequent loading of 12 N.m for both position controllers at Case (1) of parameter uncertainties.

(a) Using PFNN position tracking controller

(b) Using IACS with RFLPFNN position tracking controller

$$\varepsilon = U_{qs}^{*} - U_{qs}^{*RFLPFNNC}(E,\Theta^{*},\mu^{*},\sigma^{*}) \qquad (56)$$

where Θ^{*} is the optimal weight vector achieves the minimum approximation error and ε is the approximation error and is assumed to be bounded by $|\varepsilon| < \delta$. Θ^{*}, μ^{*} and σ^{*} are the optimal parameters of Θ, μ and σ, respectively, in the RFLPFNNC. From (56), the error equation in (55) can be rewritten as follows:

$$\begin{aligned}\dot{E} &= \Lambda E + \overline{B}_m\left(U_{qs}^{*} - \hat{U}_{qs}^{RFLPFNNC} - U_{qs}^{RC}\right)\\ &= \Lambda E + \overline{B}_m\{[U_{qs}^{*} - U_{qs}^{*RFLPFNNC}]\\ &\quad + [U_{qs}^{*RFLPFNNC} - \hat{U}_{qs}^{RFLPFNNC}] - U_{qs}^{RC}\} \qquad (57)\\ &= \Lambda E + \overline{B}_m\left(\varepsilon + \tilde{U}_{qs} - U_{qs}^{RC}\right)\\ &= \Lambda E + \overline{B}_m\left(\varepsilon + \tilde{\Theta}^{T}\Omega^{*} + \hat{\Theta}^{T}\tilde{\Omega} - U_{qs}^{RC}\right)\end{aligned}$$

where $\tilde{\Theta} = (\Theta^{*} - \hat{\Theta})$ and $\tilde{\Omega} = (\Omega^{*} - \hat{\Omega})$. The weights of the RFLPFNNC are updated online to guarantee the closed-loop stability and perfect tracking performance. To achieve this goal, the linearization technique is used to transform the nonlinear output of PRFLPFNNC into partially linear form so that the Lyapunov theorem extension can be applied. The expansion of $\tilde{\Omega}$ in Taylor series is obtained as follows:

$$\tilde{\Omega} = \begin{bmatrix}\tilde{\Omega}_1\\ \tilde{\Omega}_2\\ \vdots\\ \tilde{\Omega}_j\end{bmatrix} = \left.\begin{bmatrix}\dfrac{\partial\Omega_1}{\partial\mu}\\ \dfrac{\partial\Omega_2}{\partial\mu}\\ \vdots\\ \dfrac{\partial\Omega_j}{\partial\mu}\end{bmatrix}^{T}\right|_{\mu=\hat{\mu}}\tilde{\mu} + \left.\begin{bmatrix}\dfrac{\partial\Omega_1}{\partial\sigma}\\ \dfrac{\partial\Omega_2}{\partial\sigma}\\ \vdots\\ \dfrac{\partial\Omega_j}{\partial\sigma}\end{bmatrix}^{T}\right|_{\sigma=\hat{\sigma}}\tilde{\sigma} + O_U \qquad (58)$$

$$\equiv \Omega_{\mu}^{T}\tilde{\mu} + \Omega_{\sigma}^{T}\tilde{\sigma} + O_U$$

where $\Omega_{\mu} = \left.\left[(\partial\Omega_1/\partial\mu)\ \ (\partial\Omega_2/\partial\mu)\ \ \cdots\ \ (\partial\Omega_j/\partial\mu)\right]^{T}\right|_{\mu=\hat{\mu}}$, $\Omega_{\sigma} = \left.\left[(\partial\Omega_1/\partial\sigma)\ \ (\partial\Omega_2/\partial\sigma)\ \ \cdots\ \ (\partial\Omega_j/\partial\sigma)\right]^{T}\right|_{\sigma=\hat{\sigma}}$, $\tilde{\mu} = (\mu^{*} - \hat{\mu})$, $\tilde{\sigma} = (\sigma^{*} - \hat{\sigma})$ and O_U is a vector of higher order terms and assumed to be pounded by a positive constant. Rewriting (58), it can be obtained that

$$\Omega^* = \hat{\Omega} + \Omega_\mu^T \tilde{\mu} + \Omega_\sigma^T \tilde{\sigma} + O_U \qquad (59)$$

Assume that

$$\tilde{U}_{qs} = (U_{qs}^* - \hat{U}_{qs}) \qquad (60)$$

From (59) and (60), we can obtain

$$\begin{aligned}\tilde{U}_{qs} &= (U_{qs}^* - \hat{U}_{qs}) = \Theta^{*T}\Omega^* - \hat{\Theta}^T\hat{\Omega} \\ &= \Theta^{*T}[\hat{\Omega} + \Omega_\mu^T\tilde{\mu} + \Omega_\sigma^T\tilde{\sigma} + O_Q] - \hat{\Theta}^T\hat{\Omega} \qquad (61) \\ &= \tilde{\Theta}^T\hat{\Omega} + \hat{\Theta}^T\Omega_\mu^T\tilde{\mu} + \hat{\Theta}^T\Omega_\sigma^T\tilde{\sigma} + \Gamma\end{aligned}$$

where the uncertain term $\Gamma = \tilde{\Theta}^T\Omega_\mu^T\tilde{\mu} + \tilde{\Theta}^T\Omega_\sigma^T\tilde{\sigma} + \Theta^{*T}O_U$ is assumed to be bounded (i.e. $|\Gamma| < \rho$). According to (57) and (59)-(61), the error dynamics can be represented as

$$\begin{aligned}\dot{E} &= \Lambda E + \overline{B}_m\left(\varepsilon + \tilde{U}_{qs} - U_{qs}^{RC}\right) \\ &= \Lambda E + \overline{B}_m(\tilde{\Theta}^T\hat{\Omega} + \hat{\Theta}^T\Omega_\mu^T\tilde{\mu} + \hat{\Theta}^T\Omega_\sigma^T\tilde{\sigma} + \varepsilon + \Gamma - U_{qs}^{RC})\end{aligned} \qquad (62)$$

Theorem: Consider the IM servo drive system represented by (11), if the intelligent controller is designed as (12), in which the adaptation laws of the RFLPFNN controller are designed as (63)-(65) and the robust controller is designed as (66) with the adaptive bound estimation algorithms given in (67) and (68). As a result, the stability of the intelligent adaptive control system can be guaranteed.

$$\dot{\hat{\Theta}} = \eta_\Theta E^T P B_m \hat{\Omega}^T \qquad (63)$$

$$\dot{\hat{\mu}} = \eta_\mu E^T P B_m \hat{\Theta}^T \hat{\Omega}_\mu^T \qquad (64)$$

$$\dot{\hat{\sigma}} = \eta_\sigma E^T P B_m \hat{\Theta}^T \hat{\Omega}_\sigma^T \qquad (65)$$

$$U_{qs}^{RC} = (\hat{\delta} + \hat{\rho})\,\mathrm{sgn}(E^T P B_m) \qquad (66)$$

$$\dot{\hat{\delta}}(t) = \eta_\delta \left|E^T P B_m\right| \qquad (67)$$

$$\dot{\hat{\rho}}(t) = \eta_\rho \left|E^T P B_m\right| \qquad (68)$$

where η_Θ, η_μ, η_σ, η_δ and η_ρ are strictly positive learning rates, $\mathrm{sgn}(\cdot)$ is the sign function, $\hat{\delta}(t)$ and $\hat{\rho}(t)$ are the on-line estimated values of the adaptive bounds $\delta(t)$ and $\rho(t)$.

Proof: To minimize the error function and to derive the adaptation laws of Θ, μ, σ, δ and ρ for the intelligent adaptive control system, a Lyapunov function is defined as:

$$\begin{aligned}&V(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t) \\ &= \frac{1}{2}E^T P E + \frac{1}{2\eta_\Theta}\tilde{\Theta}^T\tilde{\Theta} + \frac{1}{2\eta_\mu}\tilde{\mu}^T\tilde{\mu} \qquad (69) \\ &\quad + \frac{1}{2\eta_\sigma}\tilde{\sigma}^T\tilde{\sigma} + \frac{1}{2\eta_\delta}\tilde{\delta}^2 + \frac{1}{2\eta_\rho}\tilde{\rho}^2\end{aligned}$$

where $\tilde{\delta} = (\hat{\delta} - \delta)$ and $\tilde{\rho} = (\hat{\rho} - \rho)$ are the estimated errors and $P \in \Re^{n\times n}$ is a symmetric positive-definite matrix, which satisfies the following Lyapunov equation [53], [54].

$$\Lambda^T P + P\Lambda = -Q \qquad (70)$$

where $Q \in \Re^{n\times n}$ is a given symmetric positive definite matrix, $Q > 0$. By taking the derivative of the Lyapunov function (69) and using (62) and (70), it is obtained that:

$$\begin{aligned}&\dot{V}(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t) \\ &= \frac{1}{2}E^T P\dot{E} + \frac{1}{2}\dot{E}^T P E + \frac{1}{\eta_\Theta}\tilde{\Theta}^T\dot{\tilde{\Theta}} + \frac{1}{\eta_\mu}\tilde{\mu}^T\dot{\tilde{\mu}} \qquad (71) \\ &\quad + \frac{1}{\eta_\sigma}\tilde{\sigma}^T\dot{\tilde{\sigma}} + \frac{1}{\eta_\delta}\tilde{\delta}\dot{\tilde{\delta}} + \frac{1}{\eta_\rho}\tilde{\rho}\dot{\tilde{\rho}}\end{aligned}$$

Substitute (62)-(68) into (71), then

$$\begin{aligned}&\dot{V}(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t) \\ &= \frac{1}{2}E^T P\dot{E} + \frac{1}{2}\dot{E}^T P E - \frac{1}{\eta_\Theta}\tilde{\Theta}^T\dot{\hat{\Theta}} - \frac{1}{\eta_\mu}\tilde{\mu}^T\dot{\hat{\mu}} \\ &\quad - \frac{1}{\eta_\sigma}\tilde{\sigma}^T\dot{\hat{\sigma}} - \frac{1}{\eta_\delta}\tilde{\delta}\dot{\hat{\delta}} - \frac{1}{\eta_\rho}\tilde{\rho}\dot{\hat{\rho}} \\ &= \frac{1}{2}E^T P\Lambda E + \frac{1}{2}E^T\Lambda^T P E + E^T P\overline{B}_m(\tilde{\Theta}^T\hat{\Omega} + \hat{\Theta}^T\Omega_\mu^T\tilde{\mu} \\ &\quad + \hat{\Theta}^T\Omega_\sigma^T\tilde{\sigma} + \varepsilon + \Gamma - U_{qs}^{RC}) - \frac{1}{\eta_\Theta}\tilde{\Theta}^T\dot{\hat{\Theta}} - \frac{1}{\eta_\mu}\tilde{\mu}^T\dot{\hat{\mu}} \\ &\quad - \frac{1}{\eta_\sigma}\tilde{\sigma}^T\dot{\hat{\sigma}} - \frac{1}{\eta_\delta}\tilde{\delta}\dot{\hat{\delta}} - \frac{1}{\eta_\rho}\tilde{\rho}\dot{\hat{\rho}} \\ &= -\frac{1}{2}E^T Q E + E^T P\overline{B}_m(\tilde{\Theta}^T\hat{\Omega} + \hat{\Theta}^T\Omega_\mu^T\tilde{\mu} \\ &\quad + \hat{\Theta}^T\Omega_\sigma^T\tilde{\sigma} + \varepsilon + \Gamma - U_{qs}^{RC}) - \frac{1}{\eta_\Theta}\tilde{\Theta}^T\dot{\hat{\Theta}} - \frac{1}{\eta_\mu}\tilde{\mu}^T\dot{\hat{\mu}} \\ &\quad - \frac{1}{\eta_\sigma}\tilde{\sigma}^T\dot{\hat{\sigma}} - \frac{1}{\eta_\delta}\tilde{\delta}\dot{\hat{\delta}} - \frac{1}{\eta_\rho}\tilde{\rho}\dot{\hat{\rho}} \\ &= -\frac{1}{2}E^T Q E + E^T P\overline{B}_m(\varepsilon + \Gamma - U_{qs}^{RC}) - (\tilde{\sigma} + \tilde{\rho})E^T P\overline{B}_m \\ &= -\frac{1}{2}E^T Q E + \left|E^T P\overline{B}_m\right|\left|\varepsilon\right| + \left|E^T P\overline{B}_m\right|\left|\Gamma\right| - (\hat{\delta} + \hat{\rho})\left|E^T P B_m\right| \\ &\quad + (\hat{\delta} - \delta^*)\left|E^T P\overline{B}_m\right| + (\hat{\rho} - \rho)\left|E^T P\overline{B}_m\right| \\ &= -\frac{1}{2}E^T Q E + \left|E^T P\overline{B}_m\right|(\left|\varepsilon\right| - \delta) + \left|E^T P\overline{B}_m\right|(\left|\Gamma\right| - \rho) \\ &\le -\frac{1}{2}E^T Q E \le 0\end{aligned} \qquad (72)$$

Since $\dot{V}(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t) \le 0$, $\dot{V}(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t)$ is a negative semi-definite function (i.e. $\dot{V}(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t) \le \ \dot{V}(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},0)$), which implies that E, $\tilde{\Theta}$, $\tilde{\mu}$, $\tilde{\sigma}$, $\tilde{\delta}$ and $\tilde{\rho}$ are bounded function. Let the function $\Xi(t) = (1/2)E^T Q E \le -\dot{V}(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t)$ and integrate the function $\Xi(t)$ with respect to time yields:

$$\int_0^t \Xi(\tau)d\tau \le V(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},0) - V(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t) \qquad (73)$$

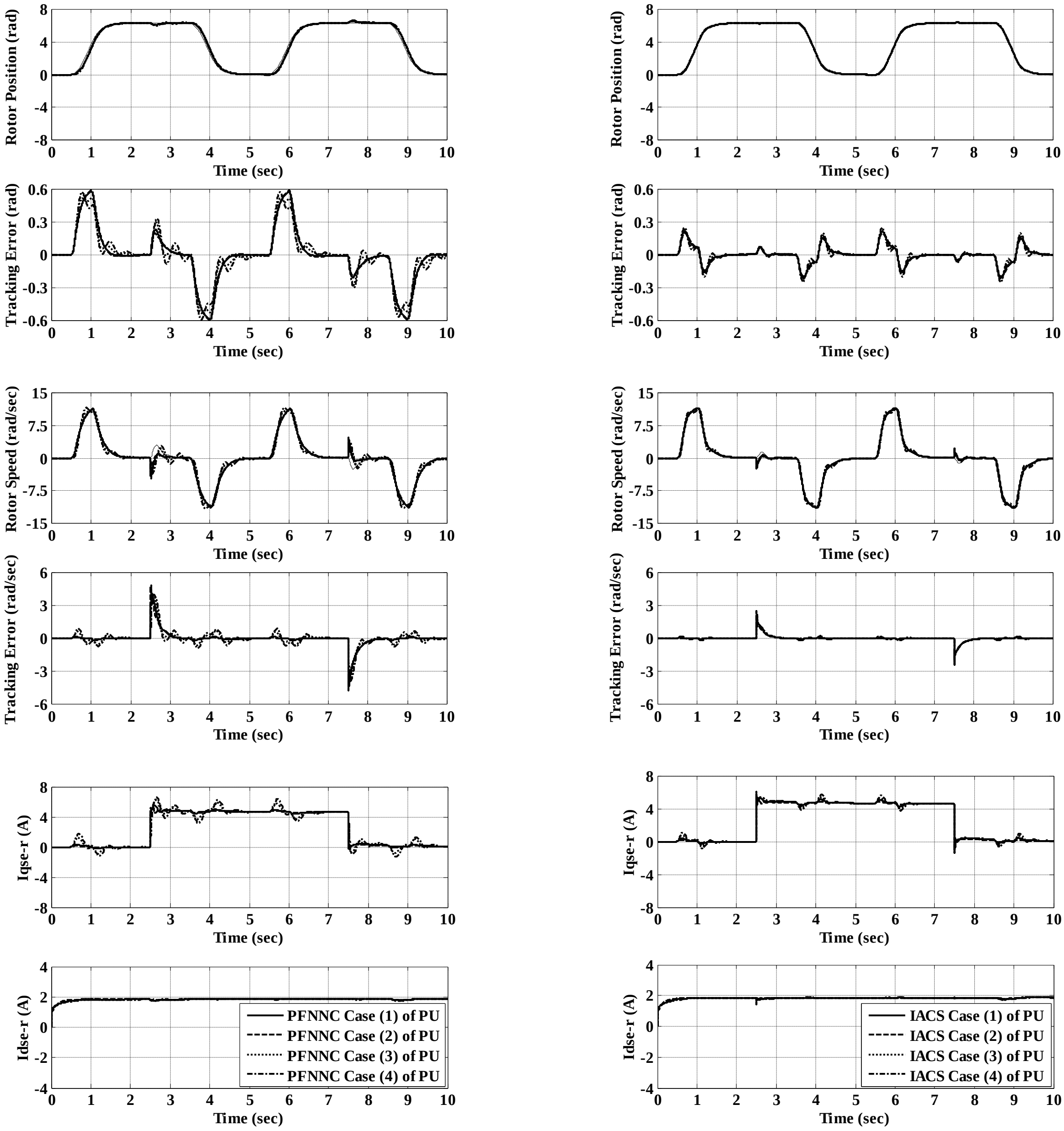
Since $V(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},0)$ is bounded and $V(E,\tilde{\Theta},\tilde{\mu},\tilde{\sigma},\tilde{\delta},\tilde{\rho},t)$ is non-increasing and bounded, the following result can be obtained:

$$\lim_{t\to\infty}\int_0^t \Xi(\tau)d\tau \le \infty \qquad (74)$$

Differentiate $\Xi(t)$ with respect to time, then

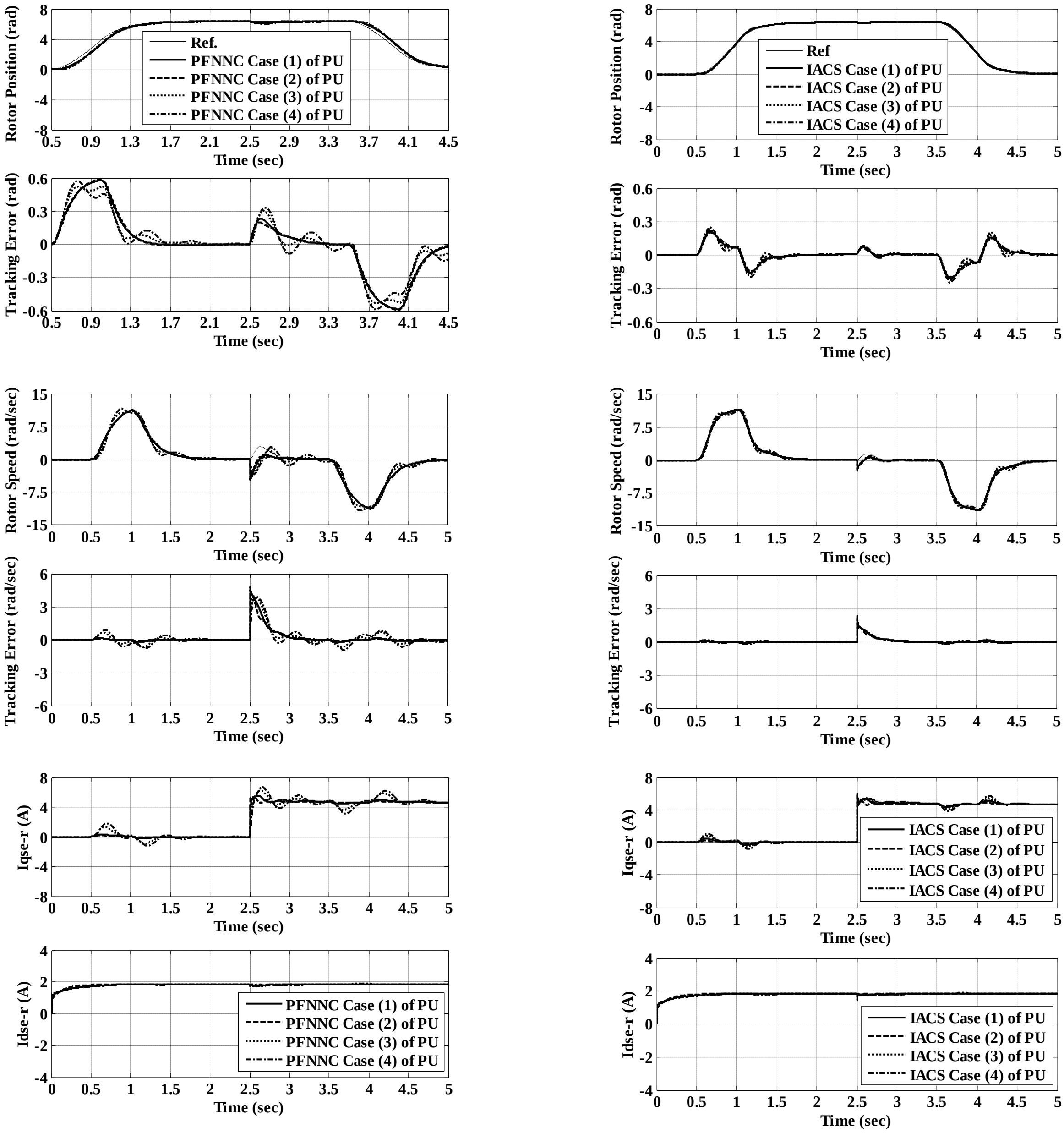
$$\dot{\Xi}(t) = E^T Q \dot{E} \qquad (75)$$

In addition, since all variables in the right hand side of (56) are bounded, it implies $\dot{E}$ is also bounded, then $\dot{\Xi}(t)$ uniformly continuous [53]. By using Barbalat's Lemma [53], [54], it can be shown that $\lim_{t\to\infty}\Xi(t)=0$. That is, $E(t)\to 0$ as $t\to\infty$. As a result, the intelligent adaptive control system is asymptotically stable. Moreover, the tracking error of the system will converges to zero according to $E(t)=0$.



(a) (b)

Fig. 6 Dynamic response of the IM servo drive system for both position controllers at different Cases (1~4) of PU
(a) Using PFNN position tracking controller (b) Using IACS with RFLPFNN position tracking controller



(a) (b)

Fig. 6 (*Continued*) Enlarge dynamic response of the IM servo drive system for both position controllers at different Cases (1~4) of PU

(a) Using PFNN position tracking controller (b) Using IACS with RFLPFNN position tracking controller

4 Numerical Simulation and Experimental Results

In order to investigate the effectiveness of the proposed tracking control scheme, the simulation and experimentation of the proposed IACS are carried out using MATLAB/SIMULINK package based on the control system shown in Figs. 1 and 4. The simulation and experimental results of the IM servo drive system are presented to verify the feasibility of the proposed IACS under various operating conditions. The threshold values, learning rates and the gains of the proposed control scheme are given as follows: D=0.35, d_{th}=0.25, η_Θ=0.75, η_μ=0.004, η_σ=0.005 η_δ=0.003, η_ρ=0.3 k_1=55 and k_2=75. All the proposed threshold values, learning rates and control gains in the IACS are chosen to achieve the superior transient control performance in the simulation and experimentation considering the limitation of control effort, the requirement of stability and the possible operating conditions. When the threshold value d_{th}=0 is set, the performance of the PFNN control system is the same as the one of the FNN control system because all the control rules in the network structure are fired at this situation. Although the computational time can be gradually decreased by increasing the threshold value, the control performance also deteriorates little by little.

Therefore, we have to compromise between the control performance and computational complexity. The control performance in the dynamic threshold value of the RFLPFNN can be approximately determined as γ=0.35 and ξ=250 in (39).

4.1 Numerical Simulation of the IM Servo Drive System

All numerical simulations are carried out using Matlab/Simulink package. The control objective is to control the position of the IM servo drive system to track the reference model trajectory. To investigate the effectiveness and robustness of the proposed control system, four simulated conditions including parameter uncertainties (PU) and external load disturbance are considered. The mechanical time constant of the IM is given by $\tau_m = (\beta_m / J_m)$.

Case 1: 1.0×τ_r, 1.0×τ_m, T_L=0–12 N.m
Case 2: 0.5×τ_r, 0.5×τ_m, T_L=0–12 N.m
Case 3: 1.5×τ_r, 2.5×τ_m, T_L=0–12 N.m
Case 4: 1.5×τ_r, 5.0×τ_m, T_L=0–12 N.m

The dynamic performance of the IM servo drive due to reference model command under subsequent loading of 12 N.m for the PFNN controller alone at Case (1) of PU including the responses of the reference model and rotor position, the tracking position error, rotor speed, the tracking speed error, d-q axis current response and adaptive signals are predicted in Fig. 5(a), respectively. On the other hand, the dynamic performance of the IM servo drive using the IACS using RFLPFNN and robust controllers is shown in Fig. 5(b) at Case (1) of PU. The disturbance rejection capabilities have been checked when a load of 12 N.m is applied to the shaft at t = 2.5 sec and removed at t = 7.5 sec. The results obtained in Fig. 5 illustrate good dynamic performances, in command tracking and load regulation performance, are realized for both position tracking controllers. Improvement of the control performance by addition the proposed IACS can be observed from the obtained results in command tracking and load regulation characteristics. From these results shown in Fig. 5, the tracking position error and tracking speed error with the PFNN controller is larger than the ones using the IACS.

To further verify the performance robustness of the proposed control schemes, four cases of PU and external load disturbance are considered, Cases (1~4), for comparison. The dynamic performance of the IM servo drive for both position controllers at all Cases of PU is predicted in Fig. 6. From the simulation results shown in Fig. 6(b), the tracking errors converges quickly and the robust control characteristics of the IM servo drive system using the proposed IACS under the occurrence of PU can be clearly observed. The proposed IACS provides a rapid and accurate response for the reference model under load changes within 0.5 sec compared with the PFNN position tracking controller which has sluggish recovery time of more than 1.0 sec at PU as shown in Fig. 6(a). Furthermore, the maximum tracking position errors at four cases of PU are approximately ±0.6 rad, for the PFNN controller. On the other hand, the ones with the IACS at four examined cases of PU are approximately constants and equal ±0.25 rad. Comparing the IACS with the PFNN controller, the tracking errors and regulation characteristics using IACS are much reduced. Therefore, the proposed IACS with RFLPFNN can yield superior control performance than the PFNN controller. As a result, the proposed IACS provides a rapid and accurate response for the reference model under load changes compared with the PFNN position tracking controller which has sluggish recovery time. Perfect tracking response and robust characteristics can still be kept with regard to PU and external disturbances as shown in Fig. 6(b). Thus, it can be verified that the proposed IACS at all cases of PU can satisfy the robustness, the accuracy requirements and is more suitable in the tracking control of the IM drive for industrial applications.

4.2 Experimentation of the IM Servo Drive System

The experimental setup for the IM servo drive system is shown in Fig. 4. A DSP control board dSPACE DS1102, which is based on a TMS320C31 and TMS320P14 DSPs, is installed in the control computer which includes multi-channels of ADC, DAC, PIO and encoder interface circuits. Digital filter and frequency multiplied by four circuits are built into the encoder interface circuits to increase the precision of the position feedback signal and coordinate transformations. The sampling rate is chosen as 200μs and hence, the carrier frequency of the PWM inverter is 5 kHz. The control interval of the position control loop is set at 1 *ms*. The current-regulated PWM VSI is implemented using MITSUBISHI intelligent power module (IPM) using IGBT components with a switching frequency of 15 kHz and driven by a six SEMIKRON IGBT drivers. The DC-link LC filter components are an inductor of iron powder core with 250μH and a polypropylene-film capacitor with 5μF. The position acquisition has been performed with a 5000 pulses/revolution incremental optical encoder. Therefore, the output of the frequency multiplier circuit is 4×5000 pulses/revolution which results high precision of the speed/position measurement.

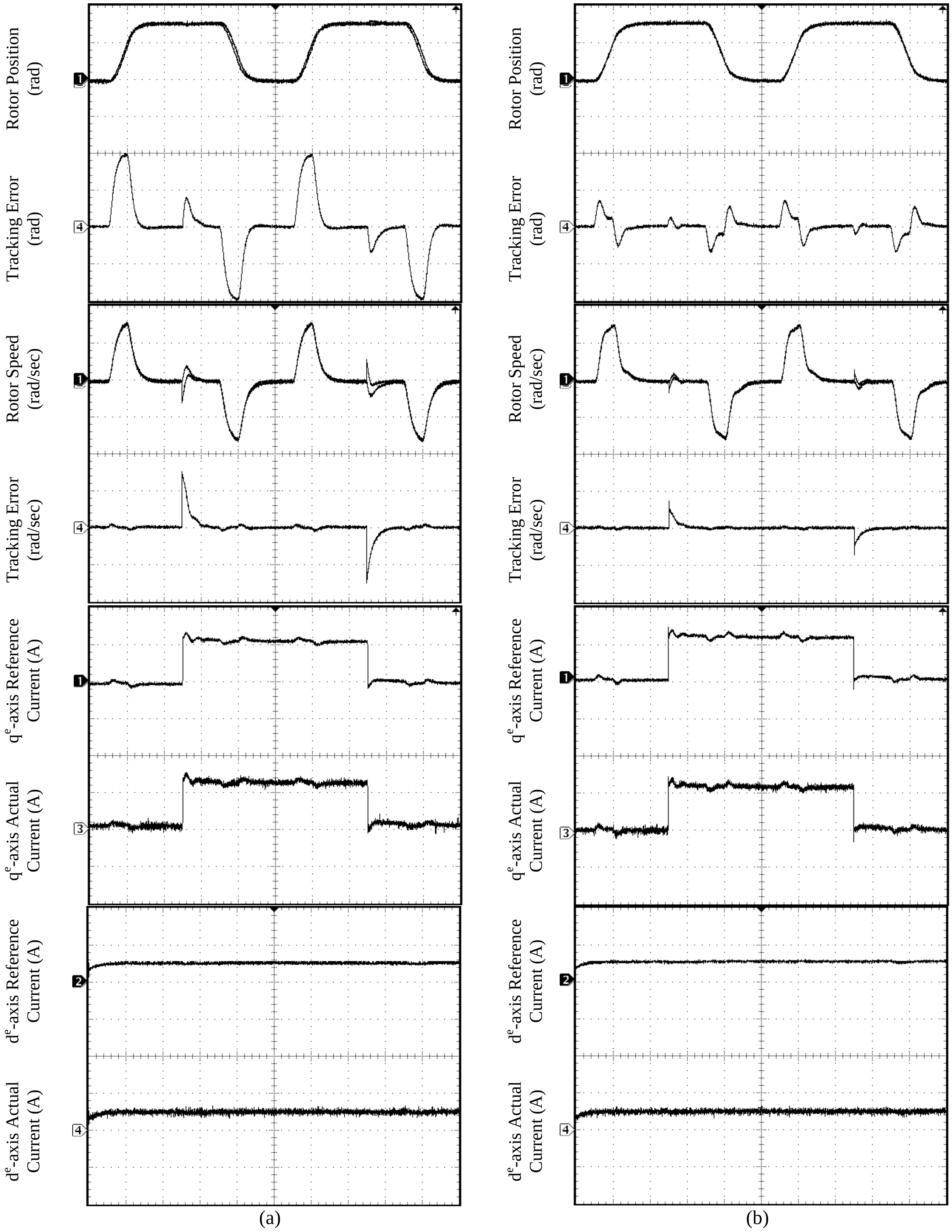
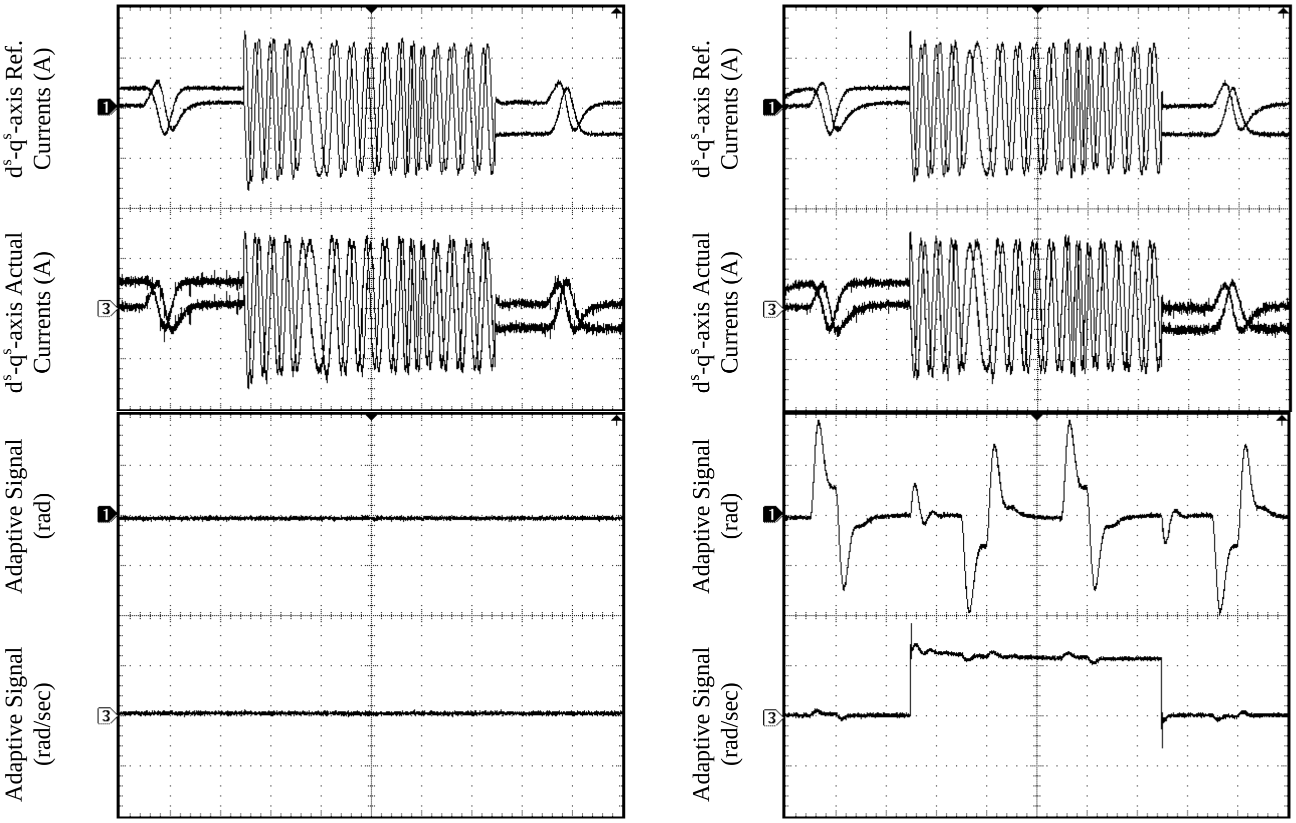


Fig. 7 Experimental results of the dynamic response of the IM servo drive for the reference position and subsequent loading of 12 N.m for both position controllers: position response 4 rad/div, speed response 7.5 (rad/sec)/div, tracking position error 0.3 rad/div, tracking speed error 3 (rad/sec)/div, q-d axis current response 4 A/div, adaptive position signal 2.5 rad/div, adaptive speed signal 10 (rad/sec)/div, time base for all traces 1 sec/div. at Case (1) of PU

(a) Using PFNN position tracking controller

(b) Using IACS with RFLPFNN position tracking controller



(a) (b)

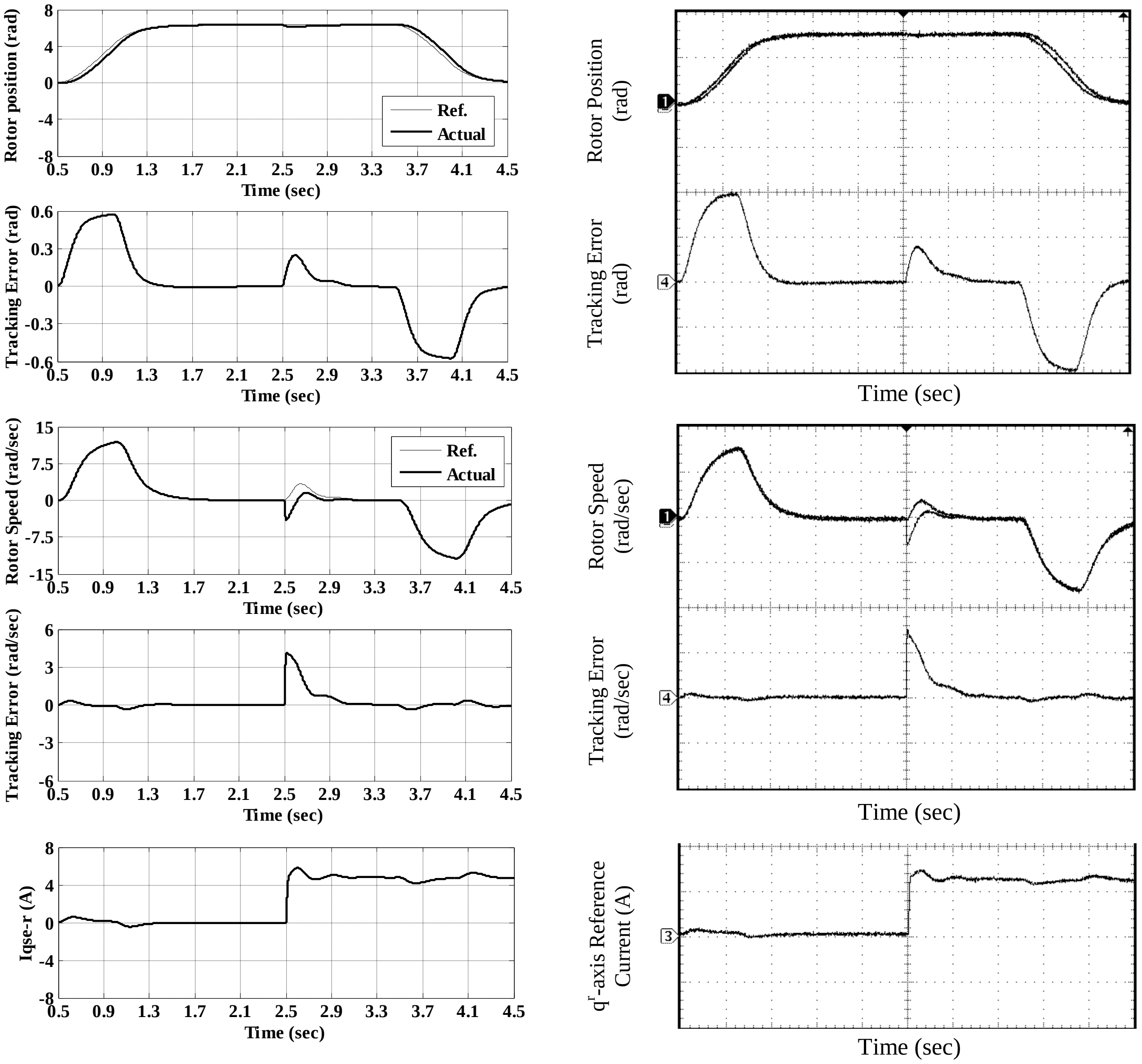
Fig. 7 (*Continued*) Experimental results of the dynamic response of the IM servo drive for the reference position and subsequent loading of 12 N.m for both position controllers: position response 4 rad/div, speed response 7.5 (rad/sec)/div, tracking position error 0.3 rad/div, tracking speed error 3 (rad/sec)/div, q-d axis current response 4 A/div, adaptive position signal 2.5 rad/div, adaptive speed signal 10 (rad/sec)/div, time base for all traces 1 sec/div. at Case (1) of PU

(a) Using PFNN position tracking controller (b) Using IACS with RFLPFNN position tracking controller

To further verify the performance of the proposed control schemes applied to the IM servo drive in practical applications, some experimental results are provided here. The experimental results of the dynamic performance for the proposed PFNN controller due to reference model command under subsequent loading of 12 N.m at Case (1) of PU including the responses of the reference model and rotor position, the tracking position error, rotor speed, the tracking speed error, d-q axis current response and adaptive control signals are predicted in Fig. 7(a), respectively. On the other hand, the experimental results of the IM servo drive using the proposed IACS is shown in Fig. 7(b) at the same conditions. Furthermore, the disturbance rejection capabilities have been checked for both position controllers. The experimental results obtained in Fig. 7 clearly illustrate good dynamic performances, in command tracking and load regulation performance, are realized for both position tracking controllers. Comparing the IACS with the PFNN controller, the tracking errors and regulation characteristics are much reduced using the proposed IACS. Therefore, the proposed IACS can yield superior control performance than the PFNN control scheme. As a result, the proposed IACS provides a rapid and accurate response for the reference model under load changes compared with the PFNN position tracking controller which has sluggish recovery time. Although the proposed IACS requires extra time to execute the computation operations and adaptive laws, it results in superior control performance than the PFNN controller. However, recent developments in microelectronics and very large scale integration (VLSI) have pushed the performance of DSPs to an unprecedented level with lower cost. Today, high-performance DSPs can be used effectively to provide flexible environments with high execution rates for advanced control schemes.

Table (2)
Performance Measures of the 2DOF I-PDC under PU of IM Servo Drive (Simulation)

Parameters Uncertainties	Tracking Errors (rad)		
	Maximum	Average	S.D.
Case (1)	0.860900	0.016470	0.237300
Case (2)	1.656000	0.020520	0.289900
Case (3)	0.824600	0.019290	0.296300
Case (4)	1.997000	0.017360	0.732000



(a) Simulation Results (b) Experimental Results

Fig. 8 Dynamic response of the IM servo drive for the reference position and subsequent loading of 12 N.m using PFNN tracking position controller at Case (1) of PU

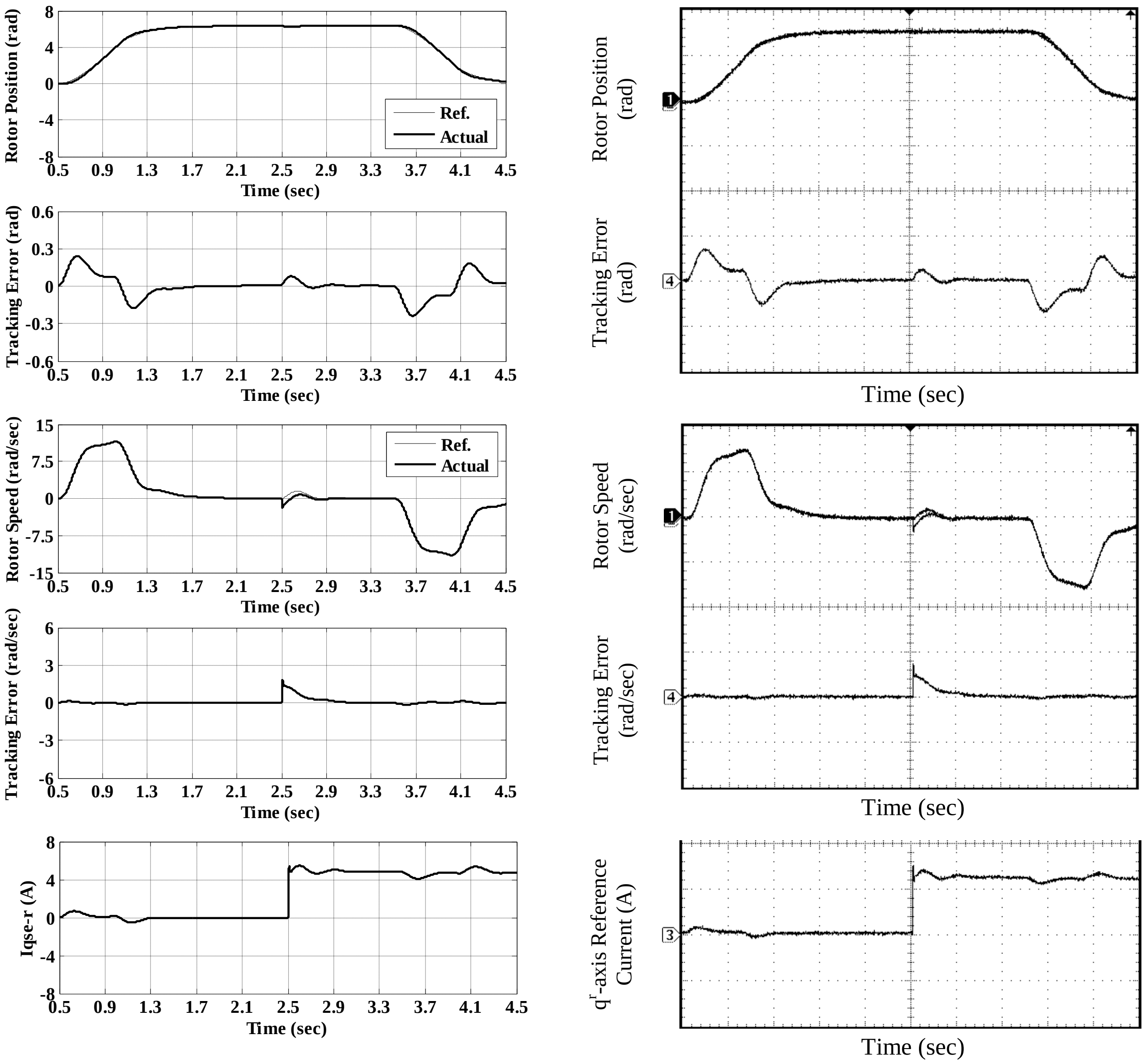
Experimental Scales: position response 4 rad/div, speed response 7.5 (rad/sec)/div, tracking position error 0.3 rad/div, tracking speed error 3 (rad/sec)/div, q- axis current response 4 A/div, time base for all traces 1 sec/div.

4.3 Comparison of Dynamic Performance in Simulation and Experimentation for the IM Servo Drive System

In this section, the dynamic performance of the induction motor drive system is compared in simulation and experimentation. The dynamic performance of the IM servo drive system using PRFNNC is shown in Fig 8 (a) and (b) for simulation and implementation, respectively, at Case (1) of PU. While, the dynamic performance using IACS is illustrated in Fig. 9 (a) and (b) for simulation and implementation, respectively, at the same operating conditions. It is clear from Figs. (8) and (9) that the experimental results are very close to the simulation results, which confirm the precise dynamic response of the IM servo drive system. In addition, it is evident that the successful practical implementation of the PFNN controller and IACS with RFLPFNN and robust controller using DSP for IM servo drive system.

Table (3)
Performance Measures of the PFNNC under PU of IM Servo Drive (Simulation)

Parameters Uncertainties	Tracking Errors (rad)		
	Maximum	Average	S.D.
Case (1)	0.582600	0.002457	0.223400
Case (2)	0.593100	0.002525	0.235800
Case (3)	0.526300	0.002186	0.225100
Case (4)	0.574800	0.002102	0.224000



(a) Simulation Results (b) Experimental Results

Fig. 9 Dynamic response of the IM servo drive for the reference position and subsequent loading of 12 N.m using IACS tracking position controller at Case (1) of PU

Experimental Scales: position response 4 rad/div, speed response 7.5 (rad/sec)/div, tracking position error 0.3 rad/div, tracking speed error 3 (rad/sec)/div, q- axis current response 4 A/div, time base for all traces 1 sec/div.

4.4 Performance Measures of the IM Servo Drive System

To measure the performance of the IM servo drive system, the maximum tracking error, TE_{max}, the average tracking error, TE_{mean} and the standard deviation of the tracking error, T_{sd}, are defined as follows [2]:

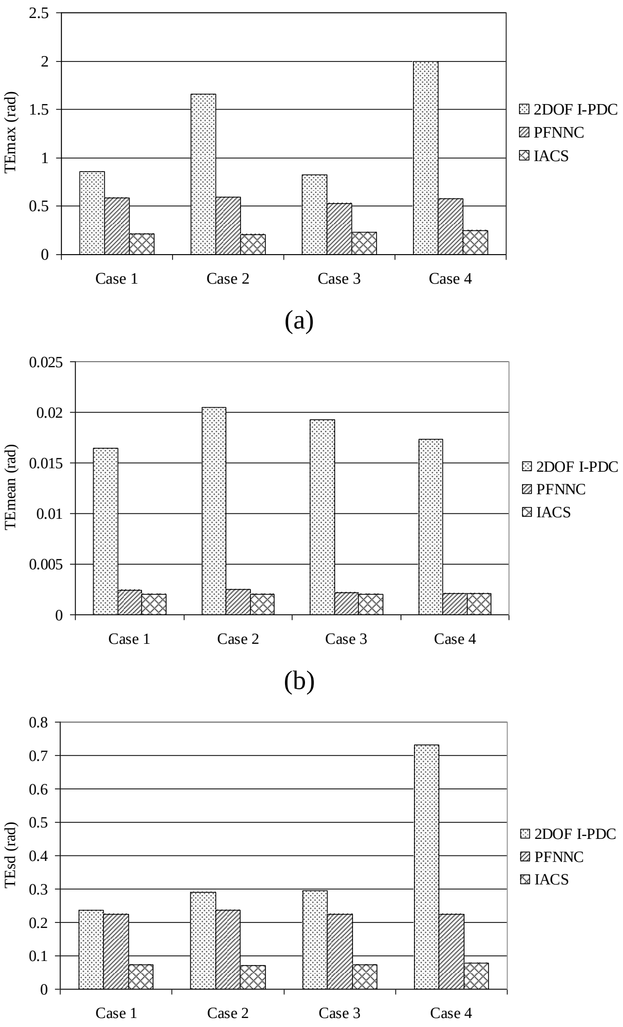
$$TE_{\max} = \max_{k} \sqrt{T(k)^2} \tag{76}$$

$$TE_{mean} = \sum_{k=1}^{n} \frac{T(k)}{n} \tag{77}$$

$$TE_{sd} = \sqrt{\sum_{k=1}^{n} \frac{(T(k) - T_{mean})^2}{n}} \tag{78}$$

where $T(k) = [\theta_r^m(k) - \theta_r(k)]$.

The performance measures can be easily demonstrated using the TE_{max} and the TE_{mean}. Moreover, the oscillation of the position tracking error can be measured using TE_{sd}. To further investigate the improvement of the proposed IACS, the performance measures of the 2DOF I-PDC, PFNNC and IACS with RFLPFNN at the four cases of PU are compared and given in Fig. 10 for simulation. The same performance measures are compared and given in Tables (2, 3 and 4) for simulation. The performance measures at case (1) of parameters uncertainties in experimentation are shown in Fig. 11. Table (5) shows the maximum tracking errors, average tracking errors and the standard deviation of the tracking errors at the nominal parameters of the IM servo drive system, case (1) in experimentation. From these results, one can easily observe that all values of TE_{max}, TE_{mean} and TE_{sd} have been successfully reduced by the proposed IACS with RFLPFNN. Therefore, the IACS possesses the best robust control characteristics and can control the IM servo drive system effectively.



(a)

(b)

(c)

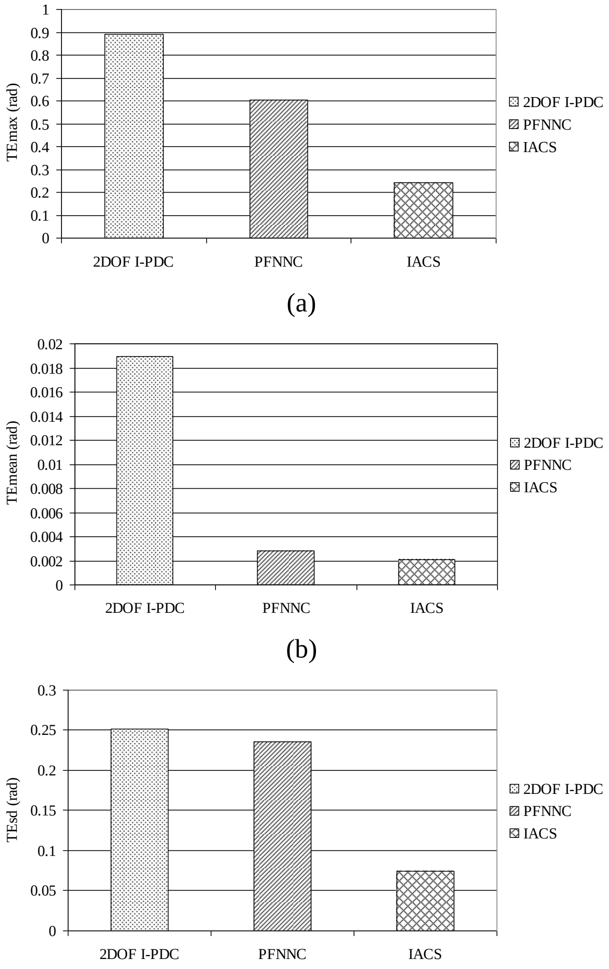
Fig. 10 Performance measures of 2DOF I-PDC, PFNNC, and IACS for IM servo drive (Simulation)
(a) TE_{max} (b) TE_{mean} (c) TE_{sd}

Table (4)
Performance Measures of the IACS with RFLPFNN under PU of IM Servo Drive (Simulation)

Parameters Uncertainties	Tracking Errors (rad)		
	Maximum	Average	S.D.
Case (1)	0.214100	0.002036	0.072060
Case (2)	0.203100	0.002030	0.071800
Case (3)	0.230900	0.002043	0.073500
Case (4)	0.250400	0.002062	0.077170

Table (5)
Performance Measures of the of PMSM Servo Drive under Nominal Parameters (Experimentation)

Controller	Tracking Errors (rad)		
	Maximum	Average	S.D.
2DOF I-PDC	0.8901	0.018960	0.25120
PFNNC	0.6032	0.002790	0.23580
IACS	0.2423	0.002095	0.07421



(a)

(b)

(c)

Fig. 11 Performance measures of 2DOF I-PDC, PFNNC and IACS for IM servo drive (Experimentation)
(a) TE_{max} (b) TE_{mean} (c) TE_{sd}

5 Conclusion

This paper proposed an IACS using RFLPFNN for IFOC-IM servo drive which guarantees the robustness in the presence of parameter uncertainties and load disturbances. The proposed IACS comprises a RFLPFNN controller and a robust controller. The RFLPFNN combines the merits of the PFNN, RFNN and FLNN. The RFLPFNN controller uses the FLNN to the consequent part of the fuzzy rules. The RFLPFNN model can automatically construct and adjust free parameters by performing online structure/parameter learning schemes concurrently. In the proposed control scheme, the

RFLPFNN controller is used as the main tracking controller to mimic an ideal control law and the robust controller is designed to compensate the difference between the ideal control law and the RFLPFNN controller. Moreover, an on-line parameter training methodology, which is derived based on the Lyapunov stability analysis and the back propagation method, is proposed to guarantee the asymptotic stability of the IACS for IM servo drive system and to increase the learning of the RFLPFNN. To relax the requirement for the bound of minimum approximation error, optimal parameter vector and higher order term in Taylor series, an adaptive control law is utilized to estimate these bounds. The theoretical bases of the proposed control scheme are derived in details. The simulated and experimental results due to reference model trajectory confirm that the proposed IACS grants robust performance and precise dynamic response regardless of load disturbances and IM parameter uncertainties. Finally, the main contribution of this paper is the successful development, application and implementation of the IACS with RFLPFNN and robust controller methodology to control the rotor position of the IM considering the existence of load disturbances and parameters uncertainties.

References

[1] R. Krishnan, *Electric motor drives: modeling, analysis, and control*, Prentice-Hall, New Jersey, 2001.

[2] F. F. M. El-Sousy, “Adaptive Dynamic Sliding-Mode Control System Using Recurrent RBFN for high-Performance Induction Motor Servo drive,” *IEEE Trans. on Industrial Informatics*, vol. 9, no. 4, pp. 1922–1936, Nov. 2013.

[3] Gi-Won Chang, G. Espinosa-Pérez, E. Mendes, and R. Ortega “Tuning Rules for the PI Gains of Field-Oriented Controllers of Induction Motors,” *IEEE Trans. Ind. Electron*., vol. 47, no. 2, pp. 592–602, June 2000.

[4] C. M. Liaw, Y. K. Chen, K. H. Chao and H. C. Chen, “Quantitative Design and Implementation of PI-D Controller with Model Following Response for Motor Drive,” *IEE Proc.-Electr. Power Appli*., vol. 145, no. 2, pp. 98-104, March 1997.

[5] H. B. Shin, J. G. Park, “Anti-Windup PID Controller With Integral State Predictor for Variable-Speed Motor Drives," *IEEE Trans. on Industrial Electronics*, vol. 59, no. 3, pp. 1509-1516, March 2012.

[6] J. Rodriguez, R.M. Kennel, J.R. Espinoza, M. Trincado, C.A. Silva, C.A. Rojas, “High-Performance Control Strategies for Electrical Drives: An Experimental Assessment,” *IEEE Trans. on Industrial Electronics*, vol. 59, no. 2, pp. 812-820, Feb. 2012.

[7] D.G. Holmes, B.P. McGrath, S.G. Parker, "Current Regulation Strategies for Vector-Controlled Induction Motor Drives ," *IEEE Trans. on Industrial Electronics*, vol. 59, no. 10, pp. 3680-3689, Oct. 2012.

[8] Yang Xia, Xinghuo Yu, and Wardina Oghanna, “Adaptive Robust Fast Control for Induction Motors ,” *IEEE Trans. Ind. Electron*., vol. 47, no. 4, pp. 854-862, August 2000.

[9] M. Nasir Uddin, and S. Woo Nam, “Development and Implementation of a Nonlinear-Controller-Based IM Drive Incorporating Iron Loss With Parameter Uncertainties,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 4, pp. 1263–1272, April 2009.

[10] A.V. Ravi Teja, C. Chakraborty, S. Maiti, Y. Hori, " A New Model Reference Adaptive Controller for Four Quadrant Vector Controlled Induction Motor Drives ," *IEEE Trans. on Industrial Electronics*, vol. 59, no. 10, pp. 3757-3767, Oct. 2012.

[11] M. Comanescu, L. Xu, and T. D. Batzel, “Decoupled Current Control of Sensorless Induction-Motor Drives by Integral Sliding Mode,” *IEEE Trans. Ind. Electron*., vol. 55, no. 11, pp. 3836-3845, Nov. 2008.

[12] R.-J.Wai, “Adaptive sliding-mode control for induction servomotor drive,” *Proc. IEE—Elect. Power Appl*., vol. 147, no. 6, pp. 553–562, November 2001.

[13] B. Veselic, B. P. Drazenovic, C. Milosavljevic, “High-Performance Position Control of Induction Motor Using Discrete-Time Sliding-Mode Control,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3809–3817, Nov. 2008.

[14] Hsin-Jang Shieh and Kuo-Kai Shyu “Nonlinear Sliding-Mode Torque Control with Adaptive Backstepping Approach for Induction Motor Drive,” *IEEE Trans. Ind. Electron*., vol. 46, no. 2, pp. 380-389, April 1999.

[15] J. Li, L. Xu, and Z. Zhang, “An Adaptive Sliding-Mode Observer for Induction Motor Sensorless Speed Control,” *IEEE Trans. Ind. Appl*., vol. 41, no. 4, pp. 1039-1046, July/August 2005.

[16] D. Traoré, F. Plestan, A. Glumineau, and J. de Leon, “Sensorless Induction Motor: High-Order Sliding-Mode Controller and Adaptive Interconnected Observer,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3818–3827, Nov. 2008.

[17] C. M. Lin, and C. F. Hsu, “Neural-Network-Based Adaptive Control for Induction Servomotor Drive System,” *IEEE Trans. Ind. Electron*., vol. 49, no. 1, pp. 115-123, Feb. 2002.

[18] R. J. Wai, “Development of Intelligent Position Control System Using Optimal Design Technique,” *IEEE Trans. Ind. Electron*., vol. 50, no. 1, pp. 218-231, Feb. 2003.

[19] P. Marino, M. Milano, and F. Vasca , “Linear Quadratic State Feedback and Robust Neural Network Estimator for Field-Oriented-Controlled Induction Motors,” *IEEE Trans. Ind. Electron*., vol. 46, no. 1, pp. 150-161, Feb. 1999.

[20] F. F. M. El-Sousy and M. M. Salem, “Simple Neuro-Controllers for Field Oriented Induction Motor Servo Drive System,” *Journal of Power Electronics*, vol. 4, no. 1, pp. 28-38, Jan. 2004.

[21] R. J. Wai and J. M. Chang, “Intelligent Control of Induction Motor Drive via Wavelet Neural Network,” *Electric Power System Research*, vol. 61, pp. 67-76, 2002.

[22] H. Rehman, “Fuzzy logic enhanced robust torque controlled induction motor drive system,” *Proc. IEE—Control Theory Appl.*, vol. 151, no. 6, pp. 754–762, Nov. 2004.

[23] R. J. Wai, R. Y. Duan, J. D. Lee, and H. H. Chang, “Wavelet Neural Network Control for Induction Motor Drive Using Sliding-Mode Design Technique,” *IEEE Trans. Ind. Electron*., vol. 50, no. 4, pp. 733-748, August 2003.

[24] R. J. Wai and F. J. Lin, “Fuzzy neural network sliding mode position controller for induction servo motor drive,” *Proc. IEE—Elect. Power Appl.*, vol. 146, no. 3, pp. 297–308, 1999.

[25] B. Castillo-Toledo, S. Di Gennaro, A. G. Loukianov, and J. Rivera, “Hybrid Control of Induction Motors via Sampled Closed Representations,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 10, pp. 3758–3771, October 2008.

[26] R. J. Wai, “Hybrid Control for Speed Sensorless Induction Motor Drive,” *IEEE Trans. Fuzzy Sys*., vol. 9, no. 1, pp. 116-138, Feb. 2001.

[27] R. J. Wai, c. M. Lin and C. F. Hsu, “Hybrid Control for Induction Servomotor Drive,” *Proc. IEE— Control Theory Appl.*, vol. 149, no. 6, pp. 555-562, Nov. 2002.

[28] R. J. Wai, H. H. Lin and F. J. Lin, “Hybrid Controller using Fuzzy Neural Networks for Identification and Control of Induction Servo Motor Drive,” *Neurocomputing*, vol. 35, pp. 91-112, 2000.

[29] C. Attaianese and G. Timasso, “H_∞ Control of Induction Motor,” *Proc. IEE—Elect. Power Appl.*, vol. 148, no. 3, pp. 272–278, May 2001.

[30] C. P. Bottura, M. F. S. Neto, and S. A. A. Filho, “Robust Speed Control of an Induction Motor: An H_∞ Control Theory Approach with Field Orientation and μ-Analysis,” *IEEE Trans. Power Electron*., vol. 15, no. 5, pp. 908-915, Sep. 2000.

[31] W. Y. Wang, Y. G. Leu, and C. C. Hsu, “Robust adaptive fuzzy-neural control of nonlinear dynamical systems using generalized projection update law and variable structure controller,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, no. 1, pp. 140–147, 2001.

[32] C. H. Wang, H. L. Liu, and T. C. Lin, “Direct adaptive fuzzy-neural control with state observer and supervisory controller for unknown nonlinear dynamical systems,” *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 1, pp. 39–49, 2002.

[33] Y. G. Leu, W. Y. Wang, and T. T. Lee, “Robust adaptive fuzzy-neural controllers for uncertain nonlinear systems,” *IEEE Trans. Robot. Automat.*, vol. 15, no. 5, pp. 805–817, 1999.

[34] J. Zhang and A. J. Morris, “Recurrent neuro-fuzzy networks for nonlinear process modeling,” *IEEE Trans. Neural Networks*, vol. 10, no. 2, pp. 313–326, 1999.

[35] C. F. Hsu and K. H. Cheng, “Recurrent Fuzzy-Neural approach for nonlinear control using dynamic structure learning scheme,” *Neurocomputing*, vol. 71, no. (16-18), pp. 3447-3459, 2008.

[36] W. Pedrycz and F.Gomide, “A Generalized Fuzzy Petri Nets,” *IEEE Transaction on Fuzzy Systems*, vol. 2, no. 4, pp. 295-301, 1994.

[37] M. Gao, M. C. Zhou, X. Haungand Z. Wu, “Fuzzy reasoning Petri Nets,” *IEEE Trans. Syst., Man, Cybern. A*, vol. 33, no. 3, pp. 314–324, 2003.

[38] R. David and H. Alla, “Petri nets for modeling of dynamic systems-A survey,” *Automatica*, vol. 30, no. 2, pp. 175–202, 1994.

[39] T. Murata, “Petri nets: Properties, analysis and applications,” *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[40] L. Xiaoou, Y. Wen, and F. Lara-Rosano, “Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework,” *IEEE Trans. Syst., Man, Cybern. C*, vol. 30, no. 4, pp. 442–450, 2000.

[41] V. R. L. Shen,, “Reinforcement learning for high-level fuzzy Petri nets,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 33, no. 2, pp. 351–362, 2003.

[42] K. Hirasawa, M. Ohbayashi, S. Sakai, and J. Hu, “Learning Petri network and its application to nonlinear system control,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, no. 6, pp. 781–789, 1998.

[43] S. I. Ahson, “Petri net models of fuzzy neural networks,” *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 6, pp. 926–932, 1995.

[44] M. Hanna, A. Buck, and R. Smith, “Fuzzy Petri nets with neural networks to model products

quality from a CNC-milling machining centre,“ *IEEE Trans. Syst., Man, Cybern. A*, vol. 26, no. 5, pp. 638–645, 1996.

[45] R. J. Wai and C. C. Chu, “Robust Petri Fuzzy-Neural-Network Control for linear induction motor drive,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 177–189, 2007.

[46] R. J. Wai and C. M. Liu, “Design of Dynamic Petri Recurrent Fuzzy Neural Network and Its Application to Path-Tracking Control of Nonholonomic Mobile Robot,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2667–2683, 2009.

[47] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, “Identification of nonlinear dynamic systems using functional link artificial neural networks,” *IEEE Trans. Syst.,Man, Cybern.*, vol. 29, no. 2, pp. 254–262, Apr. 1999.

[48] Faa-Jeng Lin, Syuan-Yi Chen, Li-Tao Teng, and Hen Chu, “Recurrent Functional-Link-Based Fuzzy Neural Network Controller With Improved Particle Swarm Optimization for a Linear Synchronous Motor Drive,” *IEEE Transactions on Magnetics*, vol. 45, no. 8, pp. 3151- 3165, August 2009.

[49] Cheng-Hung Chen, Cheng-Jian Lin, and Chin-Teng Lin, “A Functional-Link-Based Neurofuzzy Network for Nonlinear System Control,” *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 5, pp. 1362-1378,October 2008.

[50] Faa-Jeng Lin, Li-Tao Teng, Jeng-Wen Lin, and Syuan-Yi Chen, “Recurrent Functional-Link-Based Fuzzy-Neural-Network-Controlled Induction-Generator System Using Improved Particle Swarm Optimization,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 5, pp. 1557-1577, May 2009.

[51] A. Sierra, J. A. Macías, and F. Corbacho, “Evolution of functional link networks,” *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 54–65, Feb. 2001.

[52] C. T. Lin and C. S. G. Lee, *Neural Fuzzy System: A Neural-Fuzzy Synergism to Intelligent System*, Englewood cliffs, NJ: Prentice-Hall, 1996.

[53] J. J. E. Slotine and W. Li, *Applied Nonlinear Control,* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[54] K.J. Astrom, , and B. Wittenmark, *Adaptive control*, Addison-Wesley, New York, 1995.