Robust Recurrent Wavelet Interval Type-2 Fuzzy-Neural-Network Control for DSP-Based PMSM Servo Drive Systems

Fayez F. M. El-Sousy[†]

[†]Dept. of Electrical Engineering, College of Engineering, Salman bin Abdulaziz University, Al-Kharj, Saudi Arabia

Abstract

In this paper, an intelligent robust control system (IRCS) for precision tracking control of permanent-magnet synchronous motor (PMSM) servo drives is proposed. The IRCS comprises a recurrent wavelet-based interval type-2 fuzzy-neural-network controller (RWIT2FNNC), an RWIT2FNN estimator (RWIT2FNNE) and a compensated controller. The RWIT2FNNC combines the merits of a self-constructing interval type-2 fuzzy logic system, a recurrent neural network and a wavelet neural network. Moreover, it performs the structure and parameter-learning concurrently. The RWIT2FNNC is used as the main tracking controller to mimic the ideal control law (ICL) while the RWIT2FNNE is developed to approximate an unknown dynamic function including the lumped parameter uncertainty. Furthermore, the compensated controller is designed to achieve L_2 tracking performance with a desired attenuation level and to deal with uncertainties including approximation errors, optimal parameter vectors and higher order terms in the Taylor series. Moreover, the adaptive learning algorithms for the compensated controller and the RWIT2FNNE are derived by using the Lyapunov stability theorem to train the parameters of the RWIT2FNNE online. A computer simulation and an experimental system are developed to validate the effectiveness of the proposed IRCS. All of the control algorithms are implemented on a TMS320C31 DSP-based control computer. The simulation and experimental results confirm that the IRCS grants robust performance and precise response regardless of load disturbances and PMSM parameters uncertainties.

Key words: L_2 tracking performance, Lyapunov satiability theorem, Permanent-magnet synchronous motor (PMSM) servo drive, Recurrent interval type-2 fuzzy-neural-network (RIT2FNN), Wavelet neural networks

I. INTRODUCTION

Recent advancements in magnetic materials, semiconductor power devices and control theories have made permanent magnet synchronous motor (PMSM) drives play a vitally important role in motion-control applications. PMSMs are widely used in high-performance applications such as industrial robots and machine tools because of their compact size, high power density, high air-gap flux density, high torque/inertia ratio, high torque capability, high efficiency and freedom from maintenance. The overall performance of the speed and/or position control of PMSM drives depend not only on the quickness and the precision of the system response, but also on the robustness of the control strategy which has been carried out to assure the same performances if exogenous

disturbances and variations of the system parameters occur. In fact, the control of PMSM drives often necessitates the determination of machine parameters. Online variation of the parameters, which essentially depends on temperature variations, saturation and skin effects, external load disturbances and unmodeled dynamics in practical applications, can affect the PMSM servo drive performance [1]-[6]. On the other hand, a computed torque controller (CTC) is utilized to linearize the nonlinear equation by cancellation of some, or all, of the nonlinear terms such that the linear feedback controller is designed to achieve the desired closed-loop performance. However, an objection to the real-time use of such control schemes is the lack of knowledge of uncertainties [7]-[9]. Therefore, to compensate for various uncertainties and nonlinearities, a sophisticated control strategy is very important in PMSM servo drives.

Nowadays, a lot of intelligent control techniques have been developed to improve the performance of PMSM servo drives and to deal with nonlinearities and uncertainties by using fuzzy

logic, neural networks, wavelet neural networks (WNN) and/or a hybrid of these approaches [10]-[15]. The concept of incorporating fuzzy logic into a neural network (NN) to constitute fuzzy-neural-network (FNN) has grown into a popular research topic [16–25]. However, all of these analyses and implementations focus on type-1 FNN. On the other hand, a type-2 fuzzy neural network (T2FNN) consists of a type-2 fuzzy linguistic process as the antecedent part and an interval neural network as the consequent part. The interval T2FNN (IT2FNN) is a multi-layer network for the realization of type-2 fuzzy inference systems, and it can be constructed from a set of type-2 fuzzy rules. Furthermore, the IT2FNN possesses the merits of both type-2 fuzzy systems and neural networks. Therefore, it does not require mathematical models and has the ability to approximate nonlinear systems. In addition, the IT2FNN is superior to type-1 FNN in the control of complicated and highly nonlinear systems such as PMSM servo drive systems. On the other hand, there are only a few studies to analyze and simulate type-2 FNN or IT2FNN [26]-[36]. In [52], [53], at the nominal parameters of a PMSM, a two-degrees-of-freedom integral plus proportional and rate feedback (2DOF I-PD) position controller is designed and analyzed. Although the desired tracking and regulation position control can be realized by using the 2DOF I-PD position controller at the nominal PMSM parameters, the performance of the servo drive is still sensitive to parameter variations. To solve this problem, an IRCS is proposed.

In this paper, an IRCS is proposed for the identification and control of the rotor position of a PMSM servo drive. First, based on the principle of L2 tracking performance, a position tracking controller is designed and analyzed. The IRCS comprises an RWIT2FNN controller (RWIT2FNNC), an RWIT2FNN estimator (RWIT2FNNE) and a compensated controller. In the proposed control scheme, the RWIT2FNNC, which combines the merits of a self-constructing interval type-2 fuzzy logic system, a recurrent neural network and a wavelet neural network, is used as the main tracking controller to mimic the ICL. Additionally, to relax the requirement of the lumped uncertainty, an RWIT2FNNE is developed to approximate an unknown dynamic function. In addition, a compensated controller is designed to achieve L2 tracking performance with a desired attenuation level and to deal with the uncertainties including approximation errors, optimal parameter vectors, and higher order terms in the Taylor series. Moreover, the adaptive learning algorithms for the compensated controller and the RWIT2FNNE are derived using the Lyapunov stability theorem to train the parameters of the RWIT2FNNE online, so that the stability of a PMSM servo drive can be guaranteed. A computer simulation is developed and an experimental system is established for demonstration and to verify the effectiveness of the proposed IRCS for PMSM servo drives. All of the control algorithms have been implemented in a control computer based on a TMS320C31

DSP and TMS320P14 DSP control board. The dynamic performance of the PMSM servo drive has been studied under load changes and parameters uncertainties. The numerical simulations and experimental results are given to demonstrate the effectiveness of the proposed IRCS.

This paper is organized as follows. Section II presents the field–oriented control (FOC) and dynamic analysis of the PMSM servo drive. Both the problem formulation and a description of the IRCS of the PMSM servo drive are introduced. The design methodology for the compensated controller and the IRCS are given in Section III. In addition, the design procedures and adaptive learning algorithms of the proposed IRCS and the compensated controller are described in details in Section III. The validity of the design procedure and the robustness of the proposed controller are verified by means of computer simulations and experimental analysis. The control algorithms have been developed in a control computer that is based on a TMS320C31 DSP and TMS320P14 DSP DS1102 board. The dynamic performance of the PMSM 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 IV. Conclusions are introduced in Section V.

II.   MODELING OF THE PMSM AND THE DYNAMIC ANALYSIS

The voltage equations of the stator windings in the rotating reference frame can be expressed in (1) and (2). Then, using FOC and setting the d-axis current as zero, the electromagnetic torque is obtained as given in (3) and (4) [1]. The parameters of the surface-mounted PMSM are listed in Table (1)

$$V_{qs}^r = R_s i_{qs}^r + L_{ss}\frac{d}{dt}i_{qs}^r + \omega_r L_{ss}i_{ds}^r + \omega_r \lambda_m' \qquad (1)$$

$$V_{ds}^r = R_s i_{ds}^r + L_{ss}\frac{d}{dt}i_{ds}^r - \omega_r L_{ss}i_{qs}^r \qquad (2)$$

The electromagnetic torque can be expressed as:

$$T_e = (3/2).(P/2).\lambda_m i_{qs}^r = K_t i_{qs}^r \qquad (3)$$

$$T_e = J_m\left(\frac{2}{P}\right)\frac{d^2}{dt^2}\theta_r + \beta_m\left(\frac{2}{P}\right)\frac{d}{dt}\theta_r + T_L \qquad (4)$$

From (3) and (4), the motion dynamics can be simplified as:

$$\ddot{\theta}_r = -\frac{\beta_m}{J_m}.\frac{P}{2}\dot{\theta}_r + \frac{K_t}{J_m(P/2)}.i_{qs}^{r*} - \frac{P}{2}.\frac{1}{J_m}.T_L \qquad (5)$$

$$\ddot{\theta}_r = A_m\dot{\theta}_r + B_m.U(t) + D_m.T_L \qquad (6)$$

Assume that the parameters of the PMSM are well known and that the external load disturbance is absent. Rewriting (6) can now represent the model of the PMSM servo drive system.

$$\ddot{\theta}_r(t) = A_m\dot{\theta}_r(t) + B_m.U(t) \qquad (7)$$

Quantity	Symbol	Value
Nominal power	P_n	1 hp (3-phase)
Stator self inductance	L_{ss}	0.05 H
Stator resistance	R_s	1.5 Ω
Voltage constant	λ_m	0.314 V.s/rad
Number of poles	P	4
Rotor inertia	J_m	0.003 kg.m^2
Friction coefficient	β_m	0.0009 N.m/rad/sec
Nominal speed (electrical)	ω_r	377 rad/sec
Rated torque	T_e	3.6 N.m
Rated current	I	4 A
Rated voltage	V_{L-L}	208 V
Rated frequency	f	60 Hz
Torque constant	K_t	0.95 N.m/A
Resolution of the encoder	n	4×10000 p/r

By considering the dynamics in (6) with parameter variations, load disturbances and unpredictable uncertainties will give:

$$\ddot{\theta}_r(t) = (A_{mn} + \Delta A_m)\dot{\theta}_r(t) + (B_{mn} + \Delta B_m).U(t) \\ + (D_{mn} + \Delta D_m).T_L \quad (8)$$

$$\ddot{\theta}_r(t) = A_{mn}\dot{\theta}_r(t) + B_{mn}.U(t) + \Gamma(t) \quad (9)$$

where A_{mn}, B_{mn} and D_{mn} are the nominal parameters of A_m, B_m and D_m respectively. ΔA_m, ΔB_m, ΔD_m and T_L are uncertainties due to the mechanical parameters J_m and β_m, and $\Gamma(t)$ is the lumped parameter uncertainty which is defined as:

$$\Gamma(t) = \Delta A_m \dot{\theta}_r(t) + \Delta B_m.U(t) + (D_{mn} + \Delta D_m)T_L \quad (10)$$

The bound of the lumped parameter uncertainty (PU) is assumed to be given. That is:

$$|\Gamma(t)| \le K^\theta \quad (11)$$

where K^θ is a given positive constant.

III. INTELLIGENT ROBUST CONTROL SYSTEM (IRCS)

In this section, an IRCS is designed for the identification and control of the rotor position of the PMSM servo drive. The IRCS comprises an RWIT2FNNC, an RWIT2FNNE and a compensated controller. In the proposed control scheme, the RWIT2FNNC is used as the main tracking controller to mimic the ICL. Additionally, to relax the requirement of the lumped uncertainty, an RWIT2FNNE is developed to approximate an unknown dynamic function. In addition, a compensated controller is designed to achieve L_2 tracking performance with a desired attenuation level and to deal with uncertainties including approximation errors, optimal parameter vectors, and higher order terms in the Taylor series. Moreover, the adaptive learning algorithms for the compensated controller and the RWIT2FNNE are derived by using the Lyapunov stability theorem to train the parameters of the RWIT2FNNE online, so that the stability of the PMSM servo drive can be guaranteed.

The control problem is to find a control law so that the rotor position, $\theta_r(t)$, can track the desired position, $\theta_r^m(t)$. To achieve this control objective, a tracking error vector is defined as $E = [e_\theta^m \quad \dot{e}_\theta^m]^T$, where $\theta_r^m(t)$ and $\dot{\theta}_r^m(t)$ are the desired position and speed of the PMSM servo drive system; and $e_\theta^m = [\theta_r^m(t) - \theta_r(t)]$, $\dot{e}_\theta^m = [\dot{\theta}_r^m(t) - \dot{\theta}_r(t)]$, and $\ddot{e}_\theta^m = [\ddot{\theta}_r^m(t) - \ddot{\theta}_r(t)]$ denote the position, speed and acceleration errors of the PMSM servo drive system. The PMSM parameters are assumed to be precisely known and the external load torque is assumed to be measurable. The ideal control law (ICL) is designed as [51]:

$$U_{ICL}(t) = i_{qs}^{r*}(t) = B_{mn}^{-1}[\ddot{\theta}_r^m(t) - A_{mn}\dot{\theta}_r(t) - \Gamma(t) + KE] \quad (12)$$

where $K = [k_1 \quad k_2]$, in which k_1 and k_2 are positive constants. Substituting (12) into (9) results in the error dynamics, $\ddot{e}_\theta^m(t) + k_2\dot{e}_\theta^m(t) + k_1 e_\theta^m(t) = 0$.

Suppose the control gain, K, is chosen such that all of the roots of the characteristic polynomial of the error dynamics 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. The PMSM servo drive system is asymptotically stable when the control effort, U_{ICL}, is applied. However, the equation of the ideal control law is not feasible in practice because the PMSM parameters vary and this variation cannot be measured or predicted. In addition, the external load torque in a PMSM system is not known. Therefore, the control effort (12) cannot be made. Hence an IRCS is proposed to approach the ICL. The configuration of the proposed IRCS for a PMSM servo drive is shown in Fig. 1. The proposed control law is assumed to take the following form:

$$U_{qs}^*(t) = U_{qs}^{RWIT2FNNC}(t) + U_{qs}^{RWIT2FNNE}(t) + U_{qs}^{L_2}(t) \quad (13)$$

where $U_{qs}^{RWIT2FNNC}(t) = i_{qs}^{e*}(t)$ is the RWIT2FNNC, $U_{qs}^{RWIT2FNNE}(t) = \Omega(t)$ is the RWIT2FNNE and $U_{qs}^{L_2}(t)$ is the compensated controller.

A. Description of the Wavelet Bases and the Wavelet Neural Network (WNN)

The architecture of a WNN is shown in Fig. 2. This is a three-layer neural network including the input, hidden, and output layers. Each output (i.e. $H_1, H_2, \cdots$ etc.) formulated by a sub-WNN is defined as a WNN base. In the RWIT2FNN, the WNN bases do not exist in the initial state. They are generated online concurrently with the fuzzy rules using the structure learning algorithm. The WNNs are characterized by weights and wavelet bases. Each linear synaptic weight of the wavelet basis is adjustable by learning. Notably, the ordinary
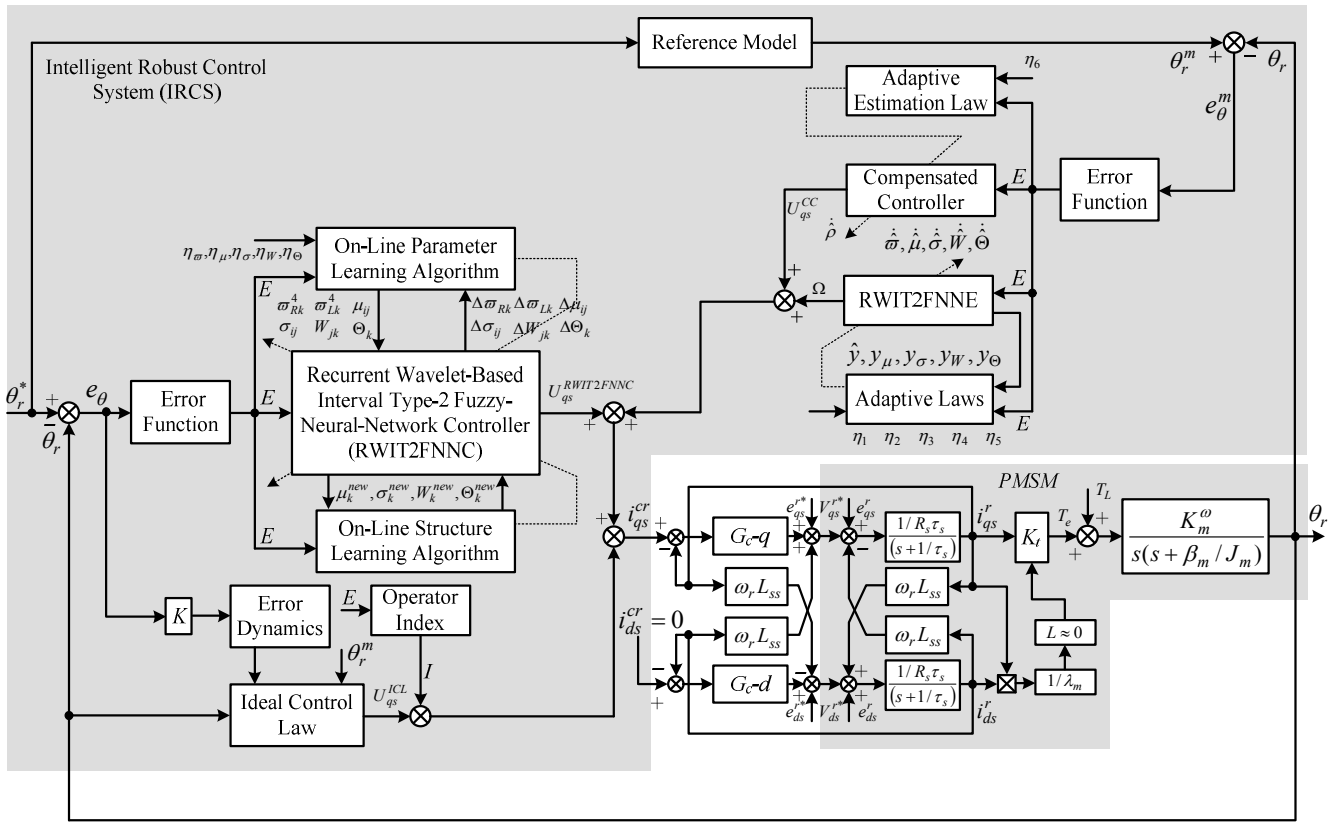
Fig. 1.   Structure of the proposed intelligent robust control system (IRCS) using RWIT2FNN for PMSM servo drive.

wavelet neural network model applications are often useful for normalizing the input vectors in the interval (0; 1) [37]-[38]. The $\phi_{a.b}(x_i)$ functions which are used as input vectors to fire up the wavelet interval are then calculated. Obviously, the value of $\phi_{a.b}$ is obtained as follows:

$$\begin{cases} \phi(x_i) = \cos(x_i) & -0.5 \le x_i \le 0.5 \\ \quad\quad 0 & (otherwise) \end{cases}, \quad \phi_{a.b} = \cos(ax_i - b)$$

(14)

where $b = 1; \ldots; a$ and $a = 1; \ldots; m.$

$$M = \sum_{a=1}^{m} \sum_{b=1}^{a} 1 \tag{15}$$

The above equation formulates the non-orthogonal wavelets in a finite range, where b denotes a shifting parameter, the maximum value of which equals the corresponding scaling parameter a. In the RWIT2FNN model, the wavelet bases do not exist in an initial state, and the amount generated by the online learning algorithm is consistent between the wavelet bases and the fuzzy rules. Obviously, a crisp value, $\phi_{a.b}$, can be obtained as follows:

$$\varphi_{a.b} = \frac{\sum_{i=1}^{n} \phi_{a.b}(x_i)}{|X|} \tag{16}$$

where $|X|$ is the number of input dimension. The final output

of the wavelet neural networks is

$$\hat{H}_j^s = \sum_{k=1}^{M} W_{jk}^s \varphi_{a.b}$$
$$= W_{j1}^s \varphi_{0.0} + W_{j2}^s \varphi_{1.0} + W_{j3}^s \varphi_{1.1} + \cdots + W_{jM}^s \varphi_{m.m}$$

(17)

where $\hat{H}_j^s$ denotes the local output of the WNN for the output, H^s, and the jth rule; the link weight, W_{jk}^s, is the output action strength associated with the sth output, the jth rule and the kth $\phi_{a.b}$; and M denotes the number of wavelet bases, which is equal to the number of existing fuzzy rules in the RWIT2FNN model.

B. Structure of the Recurrent Wavelet-Based IT2FNN

This subsection introduces the structure of the RWIT2FNN model. The RWIT2FNN integrates an interval type-2 fuzzy logic system, a recurrent neural network and a WNN. The goal of integrating the RWIT2FNN model with a WNN model is to improve the accuracy of the function approximation. The interval type-2 Gaussian MF is constructed by a type-1 Gaussian MF with an adjustable uncertain mean and an adjustable standard deviation [27-29]. Fig. 3 shows a 2-D type-2 Gaussian MF with an adjustable uncertain mean in $[\bar{\mu}, \underline{\mu}]$ and an adjustable standard

deviation, σ. It can be described as

$$\lambda_{\widetilde{A}}(x) = \exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right], \quad \mu \in [\overline{\mu}, \underline{\mu}] \qquad (18)$$

The type-2 fuzzy set has a region called the footprint of uncertainty and it is bounded by an upper MF and a lower MF [34], which are denoted as $\overline{\lambda}_{\widetilde{A}}(x)$ and $\underline{\lambda}_{\widetilde{A}}(x)$, respectively. The suggested recurrent WNN-based RIT2FNN is presented as follows:

R_j : IF

x_1^1 is $\widetilde{A}_1^j$ and ... and x_n^1 is $\widetilde{A}_n^j$ and $h_j(N)$ is F_j

THEN

$\hat{H}_j$ is $\sum\limits_{k=1}^{M} W_{jk}\varphi_{a.b}$, $h_j(N+1)$ is Θ_j and y_1 is

$[\varpi_{Rk}^4, \varpi_{Lk}^4]$ \qquad (19)

where R_j is the jth rule; h_j is the internal variable; $\hat{H}_j$ is the jth WNN base which is the jth output of the local model for rule R_j; $\widetilde{A}_1^j$ is the interval type-2 fuzzy set of the antecedent part; F_j is the output of the recurrent layer; $[\varpi_{Rk}^4, \varpi_{Lk}^4]$ is a centroid set with a membership grade of the secondary MF setting to unity, which can be called the weighting interval set, derived from the interval type-2 fuzzy sets in the consequent part [31]; y_k^3 is the output of layer 3; Θ_j and W_{jk} are the consequent part parameters for the outputs h_j and $\phi_{a.b}$, respectively; and y_o^5 is the output of the RWIT2FNN.

The architecture of the RWIT2FNN is a five-layer IT2FNN embedded with dynamic feedback connections and an WNN, as shown in Fig. 4. The basic functions and signal propagation for each layer are described as follows.

1) Layer 1: input layer: Each node i in this layer is an input node, which corresponds to one input variable. These nodes only pass the input signal to the next layer, and the input variables of the recurrent IT2FNN and WNN are the same. In this layer, the node input and output are represented as

$$net_i^1(N) = x_i^1 \qquad (20)$$

$$y_i^1(N) = f_i^1(net_i^1(N)) = net_i^1(N) \quad i = 1,2 \qquad (21)$$

$$x_1^1 = e_\theta^m(t) \quad \text{and} \quad x_2^1 = \dot{e}_\theta^m(t) \qquad (22)$$

where x_i^1 represents the ith input to the node of layer 1, and N denotes the number of iterations. The input variables are $x_1^1 = e_\theta^m = (\theta_r^m - \theta_r)$, which is the tracking position error between the desired position command $\theta_r^m(t)$ and the rotor position $\theta_r^m(t)$, and $x_2^1 = \dot{e}_\theta^m = (\dot{\theta}_r^m - \dot{\theta}_r)$, which is the tracking position error change of the rotor.

2) Layer 2: membership layer: In this layer, each node performs an interval type-2 fuzzy MF, as shown in Fig. 2. For the jth node the input and output of the membership node can be described as follows:
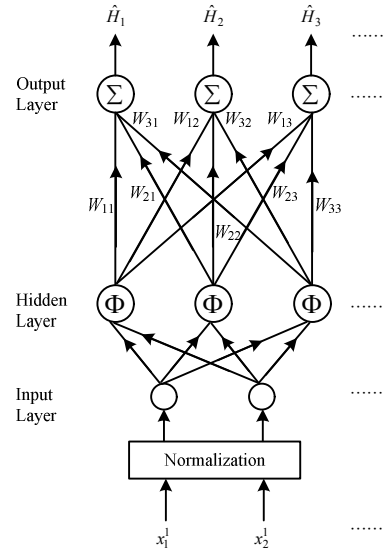

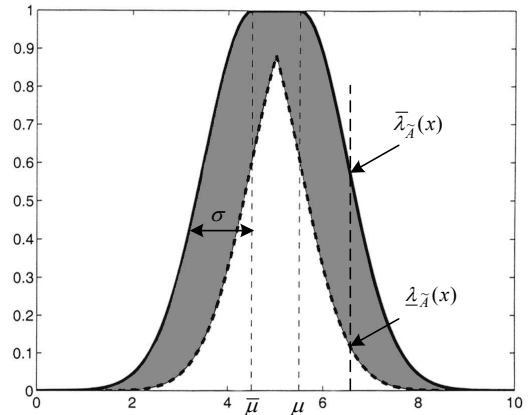Fig. 2. Wavelet network basis.

Fig. 3. Interval type-2 fuzzy set with adjustable uncertain mean and adjustable standard deviation.

$$net_j^2(N) = -(1/2)(x_i^2 - \mu_{ij})^2 / \sigma_{ij}^2 \qquad (23)$$

$$y_j^2(N) = \widetilde{A}_i^j(x_i^2) = f_j^2(net_j^2(N)) = \exp(net_j^2(N))$$

$$= \exp\left[-\frac{1}{2}\frac{\left(x_i^2 - \mu_{ij}\right)^2}{\left(\sigma_{ij}\right)^2}\right]$$

$$= \begin{cases} \overline{y}_j^2(N) & as & \mu_{ij} = \overline{\mu}_{ij} \\ \underline{y}_j^2(N) & as & \mu_{ij} = \underline{\mu}_{ij} \end{cases}, \quad j = 1,..,s \qquad (24)$$

where μ_{ij} and σ_{ij} are the mean and standard deviation of the Gaussian function in the jth term of the ith input linguistic variable x_i^2 to the node of layer 2, respectively, and s is the number of linguistic values with respect to each input node. As shown in Fig. 3, type-2 MFs can be represented as an interval bound by the upper MF $\overline{\lambda}_{\widetilde{A}}(x)$ and the lower MF

$\underline{\lambda}_{\tilde{A}}(x)$. Therefore, the output of layer 2, $y_j^2(N)$, is also represented as $[\bar{y}_j^2(N), \underline{y}_j^2(N)]$.

3) Layer 3: rule layer: This layer includes the rule layer and the recurrent layer of the RWIT2FNN and the output layer of the WNN, in which each output is defined as a WNN base. For the internal variable h_k in the recurrent layer, the following sigmoid membership function is used:

$$F_k = \frac{1}{1+\exp\{-h_k\}}, \quad k=1, 2, \ldots\ldots, m \tag{25}$$

$$h_k(N) = \sum_{k=1}^{m} u_k^3 \Theta_k \tag{26}$$

where h_k is the recurrent unit acting as the memory element, and Θ_k is the recurrent weight. Moreover, the neurons in the rule layer represent the preconditioning part of a one interval type-2 fuzzy logic rule. Thus the neuron in this layer is denoted by Π, which multiplies the incoming signals from layer 2 by the recurrent layer, and then outputs the product result, i.e., the firing strength of a rule. For the *k*th rule node:

$$net_k^3(N) = F_k \prod_j \varpi_{jk}^3 x_j^3 u_k^3(N) \tag{27}$$

$$u_k^3(N) = f_k^3(net_k^3(N)) = net_k^3(N) \tag{28}$$

Substituting (25) and (27) into (28) yields:

$$u_k^3(N) = F_k \prod_{j=1}^{n} \varpi_{jk}^3 x_j^3 u_k^3(N) = \frac{1}{1+\exp\{-h_k\}} \prod_{j=1}^{n} \varpi_{jk}^3 y_j^2 \tag{29}$$

$$u_k^3(N) = \begin{cases} \bar{u}_k^3(N) = \dfrac{1}{1+\exp\{-h_k\}} \prod_{j=1}^{n} \varpi_{jk}^3 \bar{y}_j^2 \\[2mm] \underline{u}_k^3(N) = \dfrac{1}{1+\exp\{-h_k\}} \prod_{j=1}^{n} \varpi_{jk}^3 \underline{y}_j^2 \end{cases}, \quad k=1,\ldots\ldots,n \tag{30}$$

Substituting (24) into (30) will give the output of this layer $[\bar{u}_k^3, \underline{u}_k^3]$ as follows:

$$u_k^3(N) = \begin{cases} \bar{u}_k^3(N) = \dfrac{1}{1+\exp\{-h_k\}} \prod_{j=1}^{n} \varpi_{jk}^3 \exp\left[-\dfrac{1}{2}\dfrac{(x_i^2-\bar{\mu}_{ij})^2}{(\sigma_{ij})^2}\right] \\[4mm] \underline{u}_k^3(N) = \dfrac{1}{1+\exp\{-h_k\}} \prod_{j=1}^{n} \varpi_{jk}^3 \exp\left[-\dfrac{1}{2}\dfrac{(x_i^2-\underline{\mu}_{ij})^2}{(\sigma_{ij})^2}\right] \end{cases}, \quad k=1,\ldots\ldots,n \tag{31}$$

where x_j^3 represents the *j*th input to the node of layer 3; ϖ_{jk}^3 are the weights between the membership layer and the rule layer, and are set to be equal to unity to simplify the implementation for real-time control; and *n* is the number of rules. Similar to layer 2, the output of this part of layer 3 is represented as $[\bar{u}_k^3, \underline{u}_k^3]$. In the second part of layer 3, the neuron in this layer multiplies the incoming signals, which

are $\hat{H}_k$ from the output of the WNN and u_k^3 from the output of layer 3 of the recurrent WIT2FNN part. The mathematical function of each node *k* is derived with (17) as:

$$y_k^3(N) = \hat{H}_k.u_k^3 = \left(\sum_{k=1}^{M} W_{jk}^s \varphi_{a.b}\right)u_k^3 \tag{32}$$

From (31) and (32) the following is obtained:

$$y_k^3(N) = \begin{cases} \prod\limits_{j=1}^{n} \sum\limits_{k=1}^{M} W_{jk}^s \varphi_{a.b}.(\bar{u}_k^3(N)) \\[3mm] \prod\limits_{j=1}^{n} \sum\limits_{k=1}^{M} W_{jk}^s \varphi_{a.b}.(\underline{u}_k^3(N)) \end{cases}, \quad k=1,\ldots\ldots,n \tag{33}$$

$$y_k^3(N) = \begin{cases} \prod\limits_{j=1}^{n} \dfrac{1}{1+\exp\{-h_k\}} \sum\limits_{k=1}^{M} W_{jk}^s \varphi_{a.b}.\varpi_{jk}^3 \exp\left[-\dfrac{1}{2}\dfrac{(x_i^2-\bar{\mu}_{ij})^2}{(\sigma_{ij})^2}\right] \\[4mm] \prod\limits_{j=1}^{n} \dfrac{1}{1+\exp\{-h_k\}} \sum\limits_{k=1}^{M} W_{jk}^s \varphi_{a.b}.\varpi_{jk}^3 \exp\left[-\dfrac{1}{2}\dfrac{(x_i^2-\underline{\mu}_{ij})^2}{(\sigma_{ij})^2}\right] \end{cases}, \quad k=1,\ldots\ldots,n \tag{34}$$

4) Layer 4: type-reduction layer: This layer is used to implement the type-reduction. The type-reduction is very intensive, and there exist many kinds of type-reductions, such as centroid, height, centre-of-sets and modified height. In a type-1 FLS, the height defuzzification is computationally inexpensive and gives satisfactory results. However, in a type-2 FLS, the height type-reduction does not perform as well. In this case, the centre-of-sets type-reduction does a better job. Therefore the centre-of-set type-reduction algorithm [32] is adopted in this paper. Furthermore, the process of this layer is described as follows:

$$net_l^4(N) = \frac{\sum\limits_{k=1}^{n} \varpi_k^4 y_k^3(N)}{\sum\limits_{k=1}^{n} y_k^3(N)} \tag{35}$$

$$y_l^4(N) = f_l^4(net_l^4(N)) = net_l^4(N)$$

$$= \begin{cases} y_{Rl}^4 = \dfrac{\sum\limits_{k=1}^{n} \varpi_{Rk}^4 y_{Rk}^3(N)}{\sum\limits_{k=1}^{n} y_{Rk}^3(N)} = \varpi_R^T y_R \\[4mm] y_{Ll}^4 = \dfrac{\sum\limits_{k=1}^{n} \varpi_{Lk}^4 y_{Lk}^3(N)}{\sum\limits_{k=1}^{n} y_{Lk}^3(N)} = \varpi_L^T y_L \end{cases}, \quad l=1 \tag{36}$$

where $\varpi_k^4 \in [\varpi_{Rk}^4, \varpi_{Lk}^4]$ is the centroid of the type-2 interval consequent set: $\varpi_R = [\varpi_{R1}^4 \varpi_{R2}^4 \cdots \varpi_{Rn}^4]^T$ and $\varpi_L = [\varpi_{L1}^4 \varpi_{L2}^4 \cdots \varpi_{Ln}^4]^T$.

$$y_R = \left[\dfrac{y_{R1}^3(N)}{\sum\limits_{k=1}^{n} y_{Rk}^3(N)} \quad \dfrac{y_{R2}^3(N)}{\sum\limits_{k=1}^{n} y_{Rk}^3(N)} \cdots \dfrac{y_{Rn}^3(N)}{\sum\limits_{k=1}^{n} y_{Rk}^3(N)}\right]^T \tag{37}$$

$$y_L = \left[\frac{y_{L1}^3(N)}{\sum_{k=1}^n y_{Lk}^3(N)} \quad \frac{y_{L2}^3(N)}{\sum_{k=1}^n y_{Lk}^3(N)} \cdots \frac{y_{Ln}^3(N)}{\sum_{k=1}^n y_{Lk}^3(N)} \right]^T \tag{38}$$

In order to compute $y_{Rl}^4(N)$ and $y_{Ll}^4(N)$, $[\varpi_{Rk}^4, \varpi_{Lk}^4]$ needs to be computed. This can be done by using the exact computational procedure given in [32]. The computation procedure for $y_{Rl}^4(N)$ and $y_{Ll}^4(N)$ is described briefly in the following. First, the weighting interval set $[\varpi_{Rk}^4, \varpi_{Lk}^4]$ $(k = 1,....,n)$ should be set before the computation of $y_l^4(N)$. Moreover, algorithms 1 and 2, listed below, are the type-reduction algorithms to compute $y_{Rl}^4(N)$ and $y_{Ll}^4(N)$ [32], [34], respectively.

Algorithm 1: Without a loss of generality, assume that the weighting intervals ϖ_{Rk}^4 and y_{Rl}^4 are arranged in ascending order, i.e., $\varpi_{R1}^4 \le \varpi_{R2}^4 \le \cdots \varpi_{Rn}^4$ and $y_{R1}^4 \le y_{R2}^4 \le \cdots y_{Rn}^4$.

(1) Compute y_{Rl}^4 in (36) by $y_{Rk}^3(N) = (\overline{y}_k^3(N) + \underline{y}_k^3(N))/2$ for $(k = 1,....,n)$, and let $\widetilde{y}_{Rl}^4 \cong y_{Rl}^4$.

(2) Find $R(1 \le \widetilde{R} \le n-1)$ such that $\varpi_{R\widetilde{R}}^4 \le \widetilde{y}_{Rl}^4 \le \varpi_{R(\widetilde{R}+1)}^4$.

(3) Compute y_{Rl}^4 in (36) with $y_{Rk}^3(N) = \underline{y}_k^3(N)$ for $k \le \widetilde{R}$ and with $y_{Rk}^3(N) = \overline{y}_k^3(N)$ for $k > \widetilde{R}$, and set $\widetilde{\widetilde{y}}_{Rl}^4 \cong y_{Rl}^4$.

(4) If $\widetilde{\widetilde{y}}_{Rl}^4 \cong \widetilde{y}_{Rl}^4$, then go to step (5). If $\widetilde{\widetilde{y}}_{Rl}^4 = \widetilde{y}_{Rl}^4$, then set $y_{Rl}^4 = \widetilde{\widetilde{y}}_{Rl}^4$ and stop.

(5) Set $\widetilde{y}_{Rl}^4 = \widetilde{\widetilde{y}}_{Rl}^4$ and return to step (2).

Algorithm 2: Without a loss of generality, assume that the weighting intervals ϖ_{Lk}^4 and y_{Ll}^4 are arranged in ascending order, i.e., $\varpi_{L1}^4 \le \varpi_{L2}^4 \le \cdots \varpi_{Ln}^4$ and $y_{L1}^4 \le y_{L2}^4 \le \cdots y_{Ln}^4$.

(1) Compute y_{Ll}^4 in (36) by $y_{Lk}^3(N) = (\overline{y}_k^3(N) + \underline{y}_k^3(N))/2$ for $(k = 1,....,n)$, and let $\widetilde{y}_{Ll}^4 \cong y_{Ll}^4$.

(2) Find $L(1 \le \widetilde{L} \le n-1)$ such that $\varpi_{L\widetilde{L}}^4 \le \widetilde{y}_{Ll}^4 \le \varpi_{L(\widetilde{L}+1)}^4$.

(3) Compute y_{Ll}^4 in (36) with $y_{Lk}^3(N) = \overline{y}_k^3(N)$ for $k \le \widetilde{L}$ and with $y_{Rk}^3(N) = \underline{y}_k^3(N)$ for $k > \widetilde{L}$, and set $\widetilde{\widetilde{y}}_{Ll}^4 \cong y_{Ll}^4$.

(4) If $\widetilde{\widetilde{y}}_{Ll}^4 \ne \widetilde{y}_{Ll}^4$, then go to step (5). If $\widetilde{\widetilde{y}}_{Ll}^4 \cong \widetilde{y}_{Ll}^4$, then set $y_{Ll}^4 = \widetilde{\widetilde{y}}_{Ll}^4$ and stop.

(5) Set $\widetilde{y}_{Ll}^4 = \widetilde{\widetilde{y}}_{Ll}^4$ and return to step (2).

Algorithms 1 and 2 provide a method to separate $y_{Rk}^3(N)$ and $y_{Lk}^3(N)$ into two sides by the points $\widetilde{R}$ and $\widetilde{L}$, respectively. Moreover, one side uses lower firing strengths $\underline{y}_k^3(N)$, and the other side uses upper firing strengths $\overline{y}_k^3(N)$. Therefore, (36) can be rewritten as

$$y_l^4(N) = \begin{cases} y_{Rl}^4 = \dfrac{\sum_{k=1}^n \varpi_{Rk}^4 y_{Rk}^3(N)}{\sum_{k=1}^n y_{Rk}^3(N)} \\ \quad = \dfrac{\sum_{k=1}^{\widetilde{R}} \varpi_{Rk}^4 \underline{y}_{Rk}^3(N) + \sum_{k=\widetilde{R}+1}^n \varpi_{Rk}^4 \overline{y}_{Rk}^3(N)}{\sum_{k=1}^{\widetilde{R}} \underline{y}_{Rk}^3(N) + \sum_{k=\widetilde{R}+1}^n \overline{y}_{Rk}^3(N)} \\ y_{Ll}^4 = \dfrac{\sum_{k=1}^n \varpi_{Lk}^4 y_{Lk}^3(N)}{\sum_{k=1}^n y_{Lk}^3(N)} \\ \quad = \dfrac{\sum_{k=1}^{\widetilde{L}} \varpi_{Lk}^4 \overline{y}_{Lk}^3(N) + \sum_{k=\widetilde{L}+1}^n \varpi_{Lk}^4 \underline{y}_{Lk}^3(N)}{\sum_{k=1}^{\widetilde{L}} \overline{y}_{Lk}^3(N) + \sum_{k=\widetilde{L}+1}^n \underline{y}_{Lk}^3(N)} \end{cases}, \quad l = 1 \tag{39}$$

5) Layer 5: output layer: This layer performs the linear combination of $y_{Rl}^4(N)$ and $y_{Ll}^4(N)$, i.e.,

$$y_o^5 = \frac{y_{Rl}^4 + y_{Ll}^4}{2} = U_{qs}^{*RWIT2FNNC}(t) \tag{40}$$

$$y_o^5 = \frac{1}{2}(\varpi_R^T y_R + \varpi_L^T y_L) = \frac{1}{2}\varpi^T y(x^1, \mu, \sigma, W, \Theta) \tag{41}$$

where y_o^5 is the output of the RWIT2FNN and $U_{qs}^{*RWIT2FNNC}$ is the control effort of the servo drive system. $\varpi = [\varpi_R^T \quad \varpi_L^T]^T$, $y = [y_R \quad y_L]^T$, $x^1 = [x_1^1 \quad x_2^1]$, $\mu = [\mu_{11} \cdots \mu_{1S} \quad \mu_{21} \cdots \mu_{2S}]$, $\sigma = [\sigma_{11} \cdots \sigma_{1S} \quad \sigma_{21} \cdots \sigma_{2S}]$ $W = [W_{1j} \cdots W_{Mj}]$, and $\Theta = [\Theta_1 \cdots \Theta_j]$.

C. On-Line Learning Algorithm of the RWIT2FNNC

1) Structure Learning Algorithm: The structure learning algorithm is responsible for on-line rule generation. The first task 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. Geometrically, a rule corresponds to a cluster in the input space, and the rule firing strength can be regarded as the degree to which input data belong to the cluster. Based on this concept, a previous study [39-41] used rule firing strength as a criterion for type-1 fuzzy rule generation. This idea is extended to type-2 fuzzy rule generation criteria in the RWIT2FNNC.
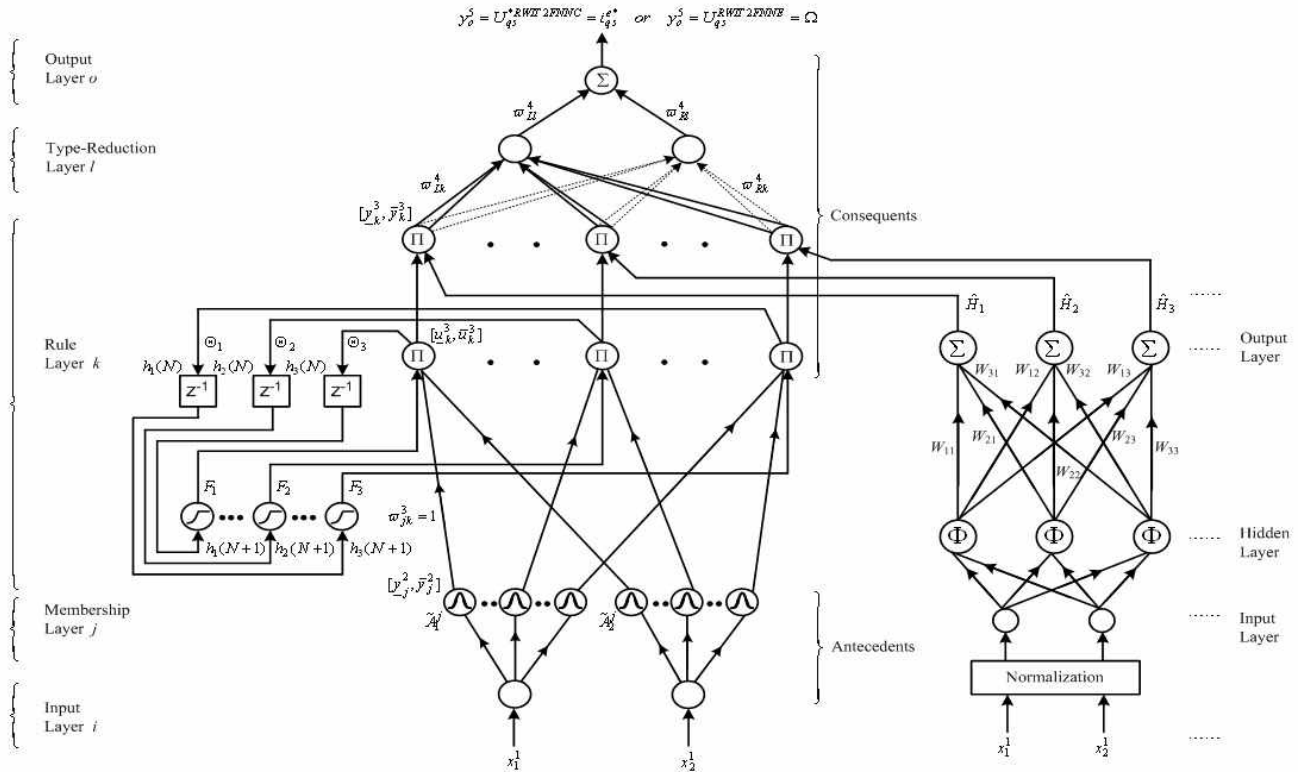
2) Parameter Learning Algorithm

Fig. 4. Structure of the recurrent wavelet-based interval type-2 fuzzy-neural-network controller (RWIT2FNNC).

The central part of the parameter-learning algorithm for the RWIT2FNNC concerns how to recursively obtain a gradient vector in which each element in the learning algorithm is defined as a derivative of an energy function with respect to a parameter of the network. This is done by means of the chain rule, and the method is generally referred to as the backpropagation learning rule, because the gradient vector is calculated in the direction opposite the flow of the output of each node. To describe the online parameter learning algorithm of the RWIT2FNNC using the supervised gradient descent method, the first the energy function is assumed as

$$E_\theta = \frac{1}{2}(\theta_r^m - \theta_r)^2 = \frac{1}{2}(e_\theta^m)^2 \qquad (42)$$

where e_θ^m is the error signal between a desired rotor position and the actual position.

Then, the update laws for the parameters in the RWIT2FNNC are described as follows:

Layer 5: During the learning process of the RWIT2FNNC, the error term to be propagated is calculated as:

$$\delta_o^5 = -\frac{\partial E_\theta}{\partial y_o^5} = \left[-\frac{\partial E_\theta}{\partial e_\theta^m} \cdot \frac{\partial e_\theta^m}{\partial y_o^5} \right] = \left[-\frac{\partial E_\theta}{\partial e_\theta^m} \cdot \frac{\partial e_\theta^m}{\partial \theta_r} \cdot \frac{\partial \theta_r}{\partial y_o^5} \right] \quad (43)$$

Layer 4: In this layer the error term needs to be calculated and propagated as

$$\delta_l^4 = -\frac{\partial E_\theta}{\partial net_l^4} = \left[-\frac{\partial E_\theta}{\partial y_o^5} \cdot \frac{\partial y_o^5}{\partial net_l^4} \right] = \frac{1}{2}\delta_o^5 \qquad (44)$$

and the weighting interval factors are updated by the following amount:

$$\Delta\varpi_{Lk}^4 = -\eta_\varpi \frac{\partial E_\theta}{\partial \varpi_{Lk}^4}$$
$$= \left[-\eta_\varpi \frac{\partial E_\theta}{\partial net_l^4} \frac{\partial net_l^4}{\partial y_{Ll}^4} \frac{\partial y_{Ll}^4}{\partial \varpi_{Lk}^4} \right] = \eta_\varpi \delta_l^4 \frac{y_k^3}{\sum_{k=1}^{n} y_k^3} \quad (45)$$

$$\Delta\varpi_{Rk}^4 = -\eta_\varpi \frac{\partial E_\theta}{\partial \varpi_{Rk}^4}$$
$$= \left[-\eta_\varpi \frac{\partial E_\theta}{\partial net_l^4} \frac{\partial net_l^4}{\partial y_{Rl}^4} \frac{\partial y_{Rl}^4}{\partial \varpi_{Rk}^4} \right] = \eta_\varpi \delta_l^4 \frac{y_k^3}{\sum_{k=1}^{n} y_k^3} \quad (46)$$

Layer 3: The error term is computed as follows:

$$\delta_k^3 = -\frac{\partial E_\theta}{\partial net_k^3} = \left[\frac{\partial E_\theta}{\partial y_o^5} \frac{\partial y_o^5}{\partial net_l^4} \frac{\partial net_l^4}{\partial y_k^3} \frac{\partial y_k^3}{\partial u_k^3} \right]\left[\frac{\partial u_k^3}{\partial net_k^3} \right]$$
$$= \delta_l^4 \frac{\varpi_{RLk}^4 \times (\sum_{k=1}^{n} \bar{y}_k^3 + \sum_{k=1}^{n} \underline{y}_k^3) - \varpi_{RLk}^4 (\bar{y}_k^3 + \underline{y}_k^3)}{\sum_{k=1}^{n} \bar{y}_k^3 + \sum_{k=1}^{n} \underline{y}_k^3} \hat{H}_k^3 \quad (47)$$

where ϖ_{RLk}^4 can be ϖ_{Rk}^4 and ϖ_{Lk}^4.

The update law of W_{jk} is:

$$\Delta W_{jk} = -\eta_W \frac{\partial E_\theta}{\partial W_{jk}} = \left[-\eta_\varpi \frac{\partial E_\theta}{\partial y_o^5} \frac{\partial y_o^5}{\partial y_k^3} \right] \left[\frac{\partial y_k^3}{\partial W_{jk}} \right] \quad (48)$$

$$= \eta_W \delta_l^4 u_k^3 \varphi_{a.b}$$

The update of Θ_k is:

$$\Delta \Theta_k = -\eta_\Theta \frac{\partial E_\theta}{\partial \Theta_k} = \left[-\eta_\Theta \frac{\partial E_\theta}{\partial u_k^3} \frac{\partial u_k^3}{\partial F_k} \frac{\partial F_k}{\partial h_k} \frac{\partial h_k}{\partial \Theta_k} \right] \quad (49)$$

$$= \eta_\Theta \delta_k^3 F_k (1 - F_k) u_k^3 (N-1) \prod_{i=1}^{n} y_j^2 (N)$$

Layer 2: The error term is computed as follows:

$$\delta_j^2 = -\frac{\partial E_\theta}{\partial net_j^2}$$

$$= \left[\frac{\partial E_\theta}{\partial y_o^5} \frac{\partial y_o^5}{\partial net_l^4} \frac{\partial net_l^4}{\partial y_k^3} \frac{\partial y_k^3}{\partial u_k^3} \frac{\partial u_k^3}{\partial net_k^3} \right] \left[\frac{\partial net_k^3}{\partial y_j^2} \frac{\partial y_j^2}{\partial net_j^2} \right] \quad (50)$$

$$= \delta_k^3 F_k$$

The update laws of the means are:

$$\Delta \underline{\mu}_{ij} = -\eta_\mu \frac{\partial E_\theta}{\partial \underline{\mu}_{ij}} = \left[-\eta_\mu \frac{\partial E_\theta}{\partial net_k^3} \frac{\partial net_k^3}{\partial y_j^2} \frac{\partial y_j^2}{\partial net_j^2} \frac{\partial net_j^2}{\partial \underline{\mu}_{ij}} \right]$$

$$= \eta_\mu \delta_j^2 \varpi_{jk}^3 \frac{(x_i^2 - \underline{\mu}_{ij})}{(\sigma_{ij})^2} \quad (51)$$

$$\Delta \overline{\mu}_{ij} = -\eta_\mu \frac{\partial E_\theta}{\partial \overline{\mu}_{ij}} = \left[-\eta_\mu \frac{\partial E_\theta}{\partial net_k^3} \frac{\partial net_k^3}{\partial y_j^2} \frac{\partial y_j^2}{\partial net_j^2} \frac{\partial net_j^2}{\partial \overline{\mu}_{ij}} \right]$$

$$= \eta_\mu \delta_j^2 \varpi_{jk}^3 \frac{(x_i^2 - \overline{\mu}_{ij})}{(\sigma_{ij})^2} \quad (52)$$

Moreover, the update law of the standard deviation is:

$$\Delta \sigma_{ij} = -\eta_\sigma \frac{\partial E_\theta}{\partial \sigma_{ij}} = \left[-\eta_\sigma \frac{\partial E_\theta}{\partial net_k^3} \frac{\partial net_k^3}{\partial y_j^2} \frac{\partial y_j^2}{\partial net_j^2} \frac{\partial net_j^2}{\partial \sigma_{ij}} \right]$$

$$= \eta_\sigma \delta_j^2 \varpi_{jk}^3 \frac{(x_i^2 - \mu_{ij})}{(\sigma_{ij})^2} \quad (53)$$

where η_ϖ, η_W, η_Θ, η_μ and η_σ are the learning-rate parameters of the weighting interval factors, the link weights of both the RIT2FNN and WNN, the feedback weights, the means and the standard deviations, respectively. To accelerate the training, the weighting interval factors, the link weights, the feedback weights, the means and the standard deviations of the membership functions are updated by including a momentum term as follows:

$$\varpi_{Lk}^4 (N+1) = \varpi_{Lk}^4 (N) + \Delta \varpi_{Lk}^4 (N) + \alpha \Delta \varpi_{Lk}^4 (N-1) \quad (54)$$

$$\varpi_{Rk}^4 (N+1) = \varpi_{Rk}^4 (N) + \Delta \varpi_{Rk}^4 (N) + \alpha \Delta \varpi_{Rk}^4 (N-1) \quad (55)$$

$$W_{jk} (N+1) = W_{jk} (N) + \Delta W_{jk} (N) + \alpha \Delta W_{jk} (N-1) \quad (56)$$

$$\Theta_k (N+1) = \Theta_k (N) + \Delta \Theta_k (N) + \alpha \Delta \Theta_k (N-1) \quad (57)$$

$$\underline{\mu}_{ij} (N+1) = \underline{\mu}_{ij} (N) + \Delta \underline{\mu}_{ij} (N) + \alpha \Delta \underline{\mu}_{ij} (N-1) \quad (58)$$

$$\overline{\mu}_{ij} (N+1) = \overline{\mu}_{ij} (N) + \Delta \overline{\mu}_{ij} (N) + \alpha \Delta \overline{\mu}_{ij} (N-1) \quad (59)$$

$$\sigma_{ij} (N+1) = \sigma_{ij} (N) + \Delta \sigma_{ij} (N) + \alpha \Delta \sigma_{ij} (N-1) \quad (60)$$

where N denotes the iteration number.

The exact calculation of the Jacobian of the system $(\partial \theta_r / \partial y_o^5)$ which is contained in (43), $(\partial E_\theta / \partial y_o^5)$, cannot be determined due to uncertainties of the servo drive system dynamics, such as parameter variations and external load disturbances. To overcome this problem and to increase the online learning rate of the network parameters, the delta adaptation law is adopted as follows:

$$\delta_o^5 = e_\theta^m + k_\theta^m \dot{e}_\theta^m \quad (61)$$

where k_θ^m is a positive constant.

D. RWIT2FNN Estimator and Compensated Controller

In this section, the RWIT2FNNE is developed to approximate an unknown dynamic function including the lumped parameter uncertainty. Furthermore, a compensated controller is designed to achieve L_2 tracking performance with desired attenuation level. Moreover, the adaptive learning algorithms for the compensated controller and the RWIT2FNNE are derived using the Lyapunov stability theorem to train the parameters of the RWIT2FNNE online.

To achieve the control objective, the tracking errors are defined as $e_\theta^m = [\theta_r^m (t) - \theta_r (t)]$, $\dot{e}_\theta^m = [\dot{\theta}_r^m (t) - \dot{\theta}_r (t)]$, and $\ddot{e}_\theta^m = [\ddot{\theta}_r^m (t) - \ddot{\theta}_r (t)]$. The tracking error function is defined as follows:

$$E(t) = \dot{e}_\theta^m (t) + \gamma e_\theta^m \quad (62)$$

where $\theta_r^m (t)$, $\dot{\theta}_r^m (t)$ and $\ddot{\theta}_r^m (t)$ are the desired position, speed and acceleration of the PMSM servo drive; and $e_\theta^m (t)$, $\dot{e}_\theta^m (t)$ and $\ddot{e}_\theta^m (t)$ denote the position, speed and acceleration errors of the PMSM servo drive. The weighting factor, γ, is used to normalize the contribution of $e_\theta^m (t)$ and $\dot{e}_\theta^m (t)$ in the error function $E(t)$. By differentiating (62), the error function becomes:

$$B_{mn}^{-1} \dot{E}(t) = B_{mn}^{-1} A_{mn} E(t) + U(t) + \Omega(t) \quad (63)$$

where the nonlinear function $\Omega(t)$ is defined as:

$$\Omega(t) = B_{mn}^{-1} \{ \Delta A_m \dot{\theta}_r (t) + \Delta B_m . U(t) + (D_{mn} + \Delta D_m) T_L + \ddot{\theta}_r^m (t) + \gamma \dot{e}_\theta^m - A_{mn} (\dot{\theta}_r (t) + \gamma e_\theta^m) \} \quad (64)$$

Taking into consideration the parameter variations of the PMSM servo drive system, $\Omega(t)$ is not only nonlinear but is also a time-varying function, consisting of commands, PMSM servo system parameters and load torque disturbances. Since the unknown function $\Omega(t)$ is very difficult to obtain in

advance in practical applications, the RWIT2FNNE is employed to estimate $\Omega(t)$ online. Furthermore, since the output of the RWIT2FNNE, $\Omega(t)$, is not able to approximate $\hat{\Omega}(t)$ accurately, a compensated controller $U_{qs}^{L_2}(t)$ is used to attenuate the approximation error. Thus the control law is defined as

$$U_{qs}^{E}(t) = \hat{\Omega}(t) + U_{qs}^{L_2}(t) \tag{65}$$

where $\Omega(t) = U_{qs}^{RWIT2FNNE}(t)$ is the output of the RWIT2FNNE.

By applying the control law (63)–(65), the closed-loop dynamics of the PMSM servo drive system can be expressed as follows:

$$B_{mn}^{-1}\dot{E}(t) = B_{mn}^{-1}A_{mn}E(t) + U(t) - U_{qs}^{RWIT2FNNC}(t)$$
$$+ \tilde{\Omega}(t) - U_{qs}^{L_2}(t) \tag{66}$$

where the approximation error $\tilde{\Omega}(t)$ is denoted as

$$\tilde{\Omega} = (\Omega - \hat{\Omega}) = \frac{1}{2}\begin{bmatrix} \varpi^{*T} y(x^1, \mu^*, \sigma^*, W^*, \Theta^*) \\ -\hat{\varpi}^{T} y(x^1, \hat{\mu}, \hat{\sigma}, \hat{W}, \hat{\Theta}) \end{bmatrix} + \varepsilon_\Omega \tag{67}$$

where ε_Ω is the minimum reconstructed error due to an insufficient number of rules; ϖ^*, μ^*, σ^*, W^* and Θ^* are the optimal parameters of ϖ, μ, σ, W and Θ; $\hat{\varpi}$, $\hat{\mu}$, $\hat{\sigma}$, $\hat{W}$ and $\hat{\Theta}$ are the estimated values of the optimal parameters (ϖ^*, μ^*, σ^*, W^* and Θ^*) as provided by the tuning algorithms that will be introduced. The approximation error $\tilde{\Omega}(t)$ in (67) is rewritten as:

$$\tilde{\Omega} = \frac{1}{2}\varpi^{*T}\tilde{y} + \frac{1}{2}\tilde{\varpi}^{T}\hat{y} + \varepsilon_\Omega \tag{68}$$

where $\tilde{\varpi} = (\varpi^* - \hat{\varpi})$ and $\tilde{y} = (y^* - \hat{y})$.

The weights of the RWIT2FNNE are updated online to make its output approximate the unknown nonlinear function $\Omega(t)$ accurately. To achieve this, the linearization technique is used to transform the nonlinear output of the RWIT2FNNE into a partially linear form so that the Lyapunov theorem extension can be applied. The expansion of $\tilde{y}$ in the Taylor series is obtained as follows:

$$\tilde{y} = \begin{bmatrix} \tilde{y}_R \\ \tilde{y}_L \end{bmatrix} = \begin{bmatrix} \dfrac{\partial y_R}{\partial \mu} \\ \dfrac{\partial y_L}{\partial \mu} \end{bmatrix}_{\mu=\hat{\mu}} (\mu^* - \hat{\mu}) + \begin{bmatrix} \dfrac{\partial y_R}{\partial \sigma} \\ \dfrac{\partial y_L}{\partial \sigma} \end{bmatrix}_{\sigma=\hat{\sigma}} (\sigma^* - \hat{\sigma})$$
$$+ \begin{bmatrix} \dfrac{\partial y_R}{\partial W} \\ \dfrac{\partial y_L}{\partial W} \end{bmatrix}_{W=\hat{W}} (W^* - \hat{W}) + \begin{bmatrix} \dfrac{\partial y_R}{\partial \Theta} \\ \dfrac{\partial y_L}{\partial \Theta} \end{bmatrix}_{\Theta=\hat{\Theta}} (\Theta^* - \hat{\Theta}) + Z \tag{69}$$
$$\equiv y_\mu^T \tilde{\mu} + y_\sigma^T \tilde{\sigma} + y_W^T \tilde{W} + y_\Theta^T \tilde{\Theta} + Z$$

where $y_\mu = \begin{bmatrix} \dfrac{\partial y_R}{\partial \mu} & \dfrac{\partial y_L}{\partial \mu} \end{bmatrix}_{\mu=\hat{\mu}}$, $y_\sigma = \begin{bmatrix} \dfrac{\partial y_R}{\partial \sigma} & \dfrac{\partial y_L}{\partial \sigma} \end{bmatrix}_{\sigma=\hat{\sigma}}$,

$y_W = \begin{bmatrix} \dfrac{\partial y_R}{\partial W} & \dfrac{\partial y_L}{\partial W} \end{bmatrix}_{W=\hat{W}}$, $y_\Theta = \begin{bmatrix} \dfrac{\partial y_R}{\partial \Theta} & \dfrac{\partial y_L}{\partial \Theta} \end{bmatrix}_{\Theta=\hat{\Theta}}$, $\tilde{\mu} = (\mu^* - \hat{\mu})$,

$\tilde{\sigma} = (\sigma^* - \hat{\sigma})$, $\tilde{W} = (W^* - \hat{W})$, $\tilde{\Theta} = (\Theta^* - \hat{\Theta})$ and Z is a vector of higher order terms and is assumed to be pounded by a positive constant. Substituting (69) into (68) will yield

$$\tilde{\Omega} = \frac{1}{2}\tilde{\varpi}^T\tilde{y} + \frac{1}{2}\hat{\varpi}^T\tilde{y} + \frac{1}{2}\tilde{\varpi}^T\hat{y} + \varepsilon_\Omega$$
$$= \frac{1}{2}\tilde{\varpi}^T(\hat{y} - y_\mu^T\hat{\mu} - y_\sigma^T\hat{\sigma} - y_W^T\hat{W} - y_\Theta^T\hat{\Theta}) \tag{70}$$
$$+ \frac{1}{2}\hat{\varpi}^T(y_\mu^T\tilde{\mu} + y_\sigma^T\tilde{\sigma} + y_W^T\tilde{W} + y_\Theta^T\tilde{\Theta}) + \Gamma$$

$$\Gamma = \frac{1}{2}\varpi^{*T}(y_\mu^T\mu^* + y_\sigma^T\sigma^* + y_W^T W^* + y_\Theta^T\Theta^* + Z)$$
$$- \frac{1}{2}\hat{\varpi}^T(y_\mu^T\mu^* + y_\sigma^T\sigma^* + y_W^T W^* + y_\Theta^T\Theta^*) + \varepsilon_\Omega \tag{71}$$

where the uncertain term Γ is assumed to be bounded by a small positive constant $|\Gamma| \leq \rho$. In order to develop the L_2 compensated controller, from (66) and (70), the error equation in can be rewritten as follows:

$$B_{mn}^{-1}\dot{E}(t) = B_{mn}^{-1}A_{mn}E(t) - U_{qs}^{L_2}(t)$$
$$+ \frac{1}{2}\tilde{\varpi}^T(\hat{y} - y_\mu^T\hat{\mu} - y_\sigma^T\hat{\sigma} - y_W^T\hat{W} - y_\Theta^T\hat{\Theta}) \tag{72}$$
$$+ \frac{1}{2}\hat{\varpi}^T(y_\mu^T\tilde{\mu} + y_\sigma^T\tilde{\sigma} + y_W^T\tilde{W} + y_\Theta^T\tilde{\Theta}) + \Gamma$$

In the case of the existence of Γ, consider the specified L_2 tracking performance that follows

$$\int_0^T (E^2(\tau)d\tau \leq \frac{1}{2}B_{mn}^{-1}E^2(0) + \frac{1}{2\eta_1}\tilde{\varpi}^T(0)\tilde{\varpi}(0)$$
$$+ \frac{1}{2\eta_2}\tilde{\mu}^T(0)\tilde{\mu}(0) + \frac{1}{2\eta_3}\tilde{\sigma}^T(0)\tilde{\sigma}(0)$$
$$+ \frac{1}{2\eta_4}\tilde{W}^T(0)\tilde{W}(0) + \frac{1}{2\eta_5}\tilde{\Theta}^T(0)\tilde{\Theta}(0) \tag{73}$$
$$+ \frac{1}{2\eta_6}\tilde{\rho}^2\int_0^T \Gamma^2(\tau)d\tau$$
$$\forall T \in [0,\infty], \quad \Gamma \in L_2[0,T]$$

where η_1, η_2, η_3, η_4 η_5 and η_6 are strictly positive learning rates; and ρ is a prescribed attenuation level. If Γ is squared integrable, so that $\int_0^\infty \Gamma^2(\tau)d\tau < \infty$, then $\lim_{t\to\infty}|E| = 0$.

If the system starts with the initial conditions $E(0) = 0$, $\tilde{\varpi}(0) = 0$, $\tilde{\mu}(0) = 0$, $\tilde{\sigma}(0) = 0$, $\tilde{W}(0) = 0$ and $\tilde{\Theta}(0) = 0$, then the L_2 tracking performance in (73) can be rewritten as

$$\sup_{\Gamma \in L_2[0,T]} \frac{\int_0^T E^2(\tau)d\tau}{\int_0^T \Gamma^2(\tau)d\tau} \le \frac{1}{2}\rho^2 \tag{74}$$

where the L_2-gain from Γ to the tracking error E must be equal to or less than $(1/2)\rho^2$. If $\rho=\infty$, this is the case of a minimum error tracking control without disturbance attenuation [42-50]. Then, the desired robust tracking performance in (73) can be achieved for a prescribed attenuation level ρ.

Theorem 1: Consider the PMSM servo drive system represented by (9), if the RWIT2FNNE control law is designed as (65), the adaptive laws of the RWIT2FNNE are designed as (75)-(79) and the compensated L_2 controller is designed as (80) with the adaptive lumped uncertainty estimation algorithm given in (81). As a result, the stability of the RWIT2FNNE system can be guaranteed.

$$\dot{\varpi} = \frac{1}{2}\eta_1(\hat{y} - y_\mu^T\hat{\mu} - y_\sigma^T\hat{\sigma} - y_W^T\hat{W} - y_\Theta^T\hat{\Theta})E(t) \tag{75}$$

$$\dot{\hat{\mu}} = \frac{1}{2}\eta_2\hat{\varpi}\ y_\mu E(t) \tag{76}$$

$$\dot{\hat{\sigma}} = \frac{1}{2}\eta_3\hat{\varpi}\ y_\sigma E(t) \tag{77}$$

$$\dot{\hat{W}} = \frac{1}{2}\eta_4\hat{W}\ y_W E(t) \tag{78}$$

$$\dot{\hat{\Theta}} = \frac{1}{2}\eta_5\hat{\Theta}\ y_\Theta E(t) \tag{79}$$

$$U_{qs}^{L_2} = \left(\frac{\rho^2+1}{2\rho^2}\right)E(t) \tag{80}$$

$$\dot{\hat{\rho}}(t) = \eta_6 E(t) \tag{81}$$

where $\dot{\hat{\rho}}(t)$ is the on-line estimated value of the hound ρ.

Proof : To minimize the error function and to derive the adaptation laws of ϖ, μ, σ, W, Θ and ρ, a Lyapunov function is defined as:

$$V_\Omega(E(t),\tilde{\varpi},\tilde{\mu},\tilde{\sigma},\tilde{W},\tilde{\Theta},\tilde{\rho}(t))$$
$$= \frac{1}{2}B_{mn}^{-1}E^2(t) + \frac{1}{2\eta_1}\tilde{\varpi}^T\tilde{\varpi} + \frac{1}{2\eta_2}\tilde{\mu}^T\tilde{\mu}$$
$$+ \frac{1}{2\eta_3}\tilde{\sigma}^T\tilde{\sigma} + \frac{1}{2\eta_4}\tilde{W}^T\tilde{W} + \frac{1}{2\eta_5}\tilde{\Theta}^T\tilde{\Theta} + \frac{1}{2\eta_6}\tilde{\rho}^2 \tag{82}$$

where $\tilde{\rho} = (\rho - \hat{\rho})$ is the estimated error. By taking the derivative of the Lyapunov function (82) and using (72) and (80) the following is obtained:

$$\dot{V}_\Omega(E(t),\tilde{\varpi},\tilde{\mu},\tilde{\sigma},\tilde{W},\tilde{\Theta},\tilde{\rho}(t))$$
$$= B_{mn}^{-1}E(t)\dot{E}(t) - \frac{1}{\eta_1}\tilde{\varpi}^T\dot{\hat{\varpi}} + \frac{1}{\eta_2}\tilde{\mu}^T\dot{\hat{\mu}} + \frac{1}{\eta_3}\tilde{\sigma}^T\dot{\hat{\sigma}}$$
$$+ \frac{1}{\eta_4}\tilde{W}^T\dot{\hat{W}} + \frac{1}{\eta_5}\tilde{\Theta}^T\dot{\hat{\Theta}} + \frac{1}{\eta_6}\tilde{\rho}(t)\dot{\hat{\rho}}(t)$$

$$= E(t)\begin{bmatrix} B_{mn}^{-1}A_{mn}E(t) - U_{qs}^{CC}(t) \\ + \frac{1}{2}\tilde{\varpi}^T(\hat{y} - y_\mu^T\hat{\mu} - y_\sigma^T\hat{\sigma} - y_W^T\hat{W} - y_\Theta^T\hat{\Theta}) \\ + \frac{1}{2}\hat{\varpi}^T(y_\mu^T\tilde{\mu} + y_\sigma^T\tilde{\sigma} + y_W^T\tilde{W} + y_\Theta^T\tilde{\Theta}) + \Gamma \end{bmatrix}$$
$$- \frac{1}{\eta_1}\tilde{\varpi}^T\dot{\hat{\varpi}} + \frac{1}{\eta_2}\tilde{\mu}^T\dot{\hat{\mu}} + \frac{1}{\eta_3}\tilde{\sigma}^T\dot{\hat{\sigma}} + \frac{1}{\eta_4}\tilde{W}^T\dot{\hat{W}}$$
$$+ \frac{1}{\eta_5}\tilde{\Theta}^T\dot{\hat{\Theta}} + \frac{1}{\eta_6}\tilde{\rho}(t)\dot{\hat{\rho}}(t) \tag{83}$$

If the adaptive update laws of the RWIT2FNNE are chosen as (75)–(79) and a robust L_2 control is designed as (80) with the adaptive bound algorithm given in (81), then (83) can be rewritten as follows:

$$\dot{V}_\Omega(E(t),\tilde{\varpi},\tilde{\mu},\tilde{\sigma},\tilde{W},\tilde{\Theta},\tilde{\rho}(t))$$

$$= E(t)\begin{bmatrix} B_{mn}^{-1}A_{mn}E(t) - \left(\frac{\rho^2+1}{2\rho^2}\right)E(t) \\ + \frac{1}{2}\tilde{\varpi}^T(\hat{y} - y_\mu^T\hat{\mu} - y_\sigma^T\hat{\sigma} - y_W^T\hat{W} - y_\Theta^T\hat{\Theta}) \\ + \frac{1}{2}\hat{\varpi}^T(y_\mu^T\tilde{\mu} + y_\sigma^T\tilde{\sigma} + y_W^T\tilde{W} + y_\Theta^T\tilde{\Theta}) + \Gamma \end{bmatrix}$$
$$- \frac{1}{2}\hat{\varpi}^T(\hat{y} - y_\mu^T\hat{\mu} - y_\sigma^T\hat{\sigma} - y_W^T\hat{W} - y_\Theta^T\hat{\Theta})E(t)$$
$$- \frac{1}{2}\hat{\varpi}\ y_\mu^T E(t) - \frac{1}{2}\hat{\varpi}\ y_\sigma^T E(t) - \frac{1}{2}\hat{W}\ y_W^T E(t)$$
$$- \frac{1}{2}\hat{\Theta}\ y_\Theta^T E(t) + \frac{1}{2}\tilde{\rho}(t)E(t)$$

$$= E(t)\left[B_{mn}^{-1}A_{mn}E(t) - \left(\frac{\rho^2+1}{2\rho^2}\right)E(t) + \Gamma\right] + \frac{1}{2}\tilde{\rho}(t)E(t)$$

$$= E(t)B_{mn}^{-1}A_{mn}E(t) - \frac{1}{2}\left(\frac{E(t)}{\rho} - \rho\ \Gamma\right)^2 + \frac{1}{2}\tilde{\rho}(t)E(t)$$
$$+ \frac{1}{2}\rho^2\Gamma^2 \le E(t)B_{mn}^{-1}A_{mn}E(t) + \frac{1}{2}\rho^2\Gamma^2 \tag{84}$$

Integrating (73) from $t = 0$ to $t = T$ yields:

$$V_\Omega(E(T),\varpi,\mu,\sigma,W,\Theta,\rho(T)) - V_\Omega(E(0),\varpi,\mu,\sigma,W,\Theta,\rho(0))$$
$$\le -\frac{1}{2}\int_0^T E^2(\tau)d\tau + \frac{1}{2}\rho^2\int_0^T \Gamma^2(\tau)d\tau \tag{85}$$

Since $V_\Omega(T) \ge 0$, the above inequality (85) implies the following inequality:

$$\frac{1}{2}\int_0^T E^2(\tau)d\tau \le V_\Omega(E(0),\varpi,\mu,\sigma,W,\Theta,\rho(0))$$

$$+\frac{1}{2}\rho^2\int_0^T\Gamma^2(\tau)d\tau \tag{86}$$

From (70) and (75), the above inequality is equivalent to the following:

$$\int_0^T(E^2(\tau)d\tau \le \frac{1}{2}B_{mn}^{-1}E^2(0)+\frac{1}{2\eta_1}\widetilde{\varpi}^T(0)\widetilde{\varpi}(0)$$

$$+\frac{1}{2\eta_2}\widetilde{\mu}^T(0)\widetilde{\mu}(0)+\frac{1}{2\eta_3}\widetilde{\sigma}^T(0)\widetilde{\sigma}(0)$$

$$+\frac{1}{2\eta_4}\widetilde{W}^T(0)\widetilde{W}(0)+\frac{1}{2\eta_5}\widetilde{\Theta}^T(0)\widetilde{\Theta}(0) \tag{87}$$

$$+\frac{1}{2\eta_6}\widetilde{\rho}^2\int_0^T\Gamma^2(\tau)d\tau$$

This is (73).

Since $\dot{V}_\Omega(E(t),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(t))\le 0$ and $\dot{V}_\Omega(E(t),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(t))$ is a negative semi-definite function i.e. $V_\Omega(E(t),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(t))\le V_\Omega(E(0),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(0))$, it can be implied that $E,\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta}$ and $\widetilde{\rho}$ are bounded functions. Let the function $\Xi(t)=\frac{1}{2}B_{mn}^{-1}E^2\le -\dot{V}_\Omega(E(t),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(t))$ and integrate the function $\Xi(t)$ with respect to time. This yields:

$$\int_0^t\Xi(\tau)d\tau=\frac{1}{2}B_{mn}^{-1}E^2\le V_\Omega(E(0),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(t))$$

$$-V_\Omega(E(t),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(t)) \tag{88}$$

Since $V_\Omega(E(0),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(0))$ is bounded and $V_\Omega(E(t),\widetilde{\varpi},\widetilde{\mu},\widetilde{\sigma},\widetilde{W},\widetilde{\Theta},\widetilde{\rho}(t))$ is non-increasing and bounded, the following result can be obtained, $\lim_{t\to\infty}\int_0^t\Xi(\tau)d\tau\le\infty$. In addition, since $\dot{\Xi}(t)$ is bounded by Barbalat's Lemma [42-43], it can be shown that $\lim_{t\to\infty}\Xi(t)=0$. That is, $E\to 0$ as $t\to\infty$. As a result, the stability of the proposed RWIT2FNNE and the compensated control system can be guaranteed.

IV. NUMERICAL SIMULATION AND EXPERIMENTAL RESULTS

In order to investigate the effectiveness of the proposed tracking control scheme, simulations and experiments on the proposed IRCS and an ideal controller are carried out using the Matlab/Simulink package based on the control system shown in Figs. 1 and 5. A DSP control board dSPACE DS1102, which is based on TMS320C31 and TMS320P14 DSPs, is installed in the control computer which includes multi-channels of ADC, DAC, PIO and encoder interface circuits. The digital filter and frequency multiplied by four
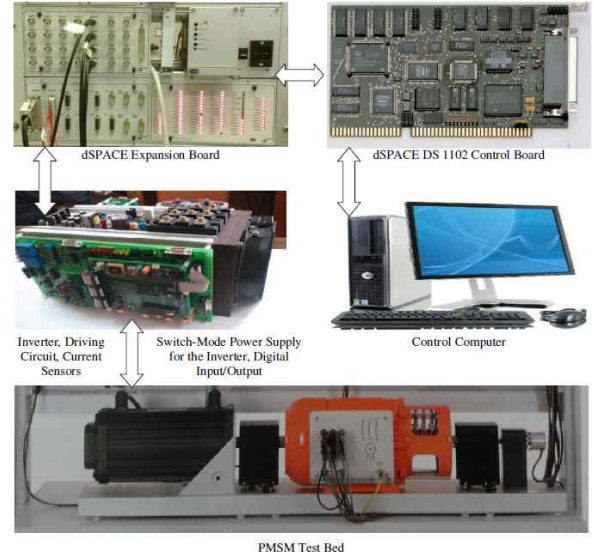
(a) Experimental setup.
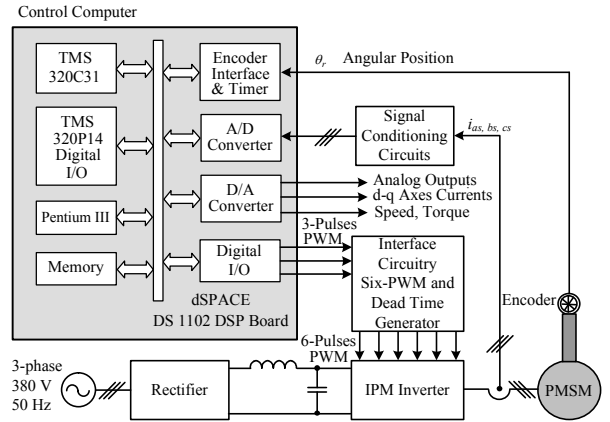
(b) Block diagram of the proposed DSP-based control system.

Fig. 5. DSP-based intelligent robust control system (IRCS) using RWIT2FNN for PMSM servo drive.

circuits are built into the encoder interface circuits to increase the precision of the speed and the position feedback signals and the coordinate transformations. The sampling rate is chosen as $200\mu s$ 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 a Mitsubishi intelligent power module (IPM) using IGBTs with a rating of 50A, 1200V and a switching frequency of 15 kHz and it is driven by six Semikron IGBT drivers. The DC-link LC filter components are an inductor with an iron powder core with 250μH and a polypropylene-film capacitor with 5μF. The speed acquisition has been performed with a 10,000 pulse/revolution incremental optical encoder. Therefore, the output of the frequency multiplier circuit is 40,000 pulses/revolution which results in high precision of the speed/position measurement.

The selection of the learning rate parameters in the online

learning algorithm of the proposed RWIT2FNN has a significant effect on the control performance since a deteriorated dynamic response results from an inappropriate selection of the learning rates. Although optimal learning rates can be obtained online by using the genetic algorithm (GA) or the particle swarm optimization (PSO) algorithm, these optimization techniques result in a high computational burden. Therefore, in the simulations and experiments the

learning rate parameters of the weighting interval factors, the link weights of both the RIT2FNN and the WNN, the feedback weights, the means and the standard deviations, respectively, η_{ϖ}, η_{W}, η_{Θ}, η_{μ} and η_{σ} are obtained by trial and error to achieve the best dynamic performance of the PMSM servo drive system, while considering the requirements of the convergence of the tracking error.
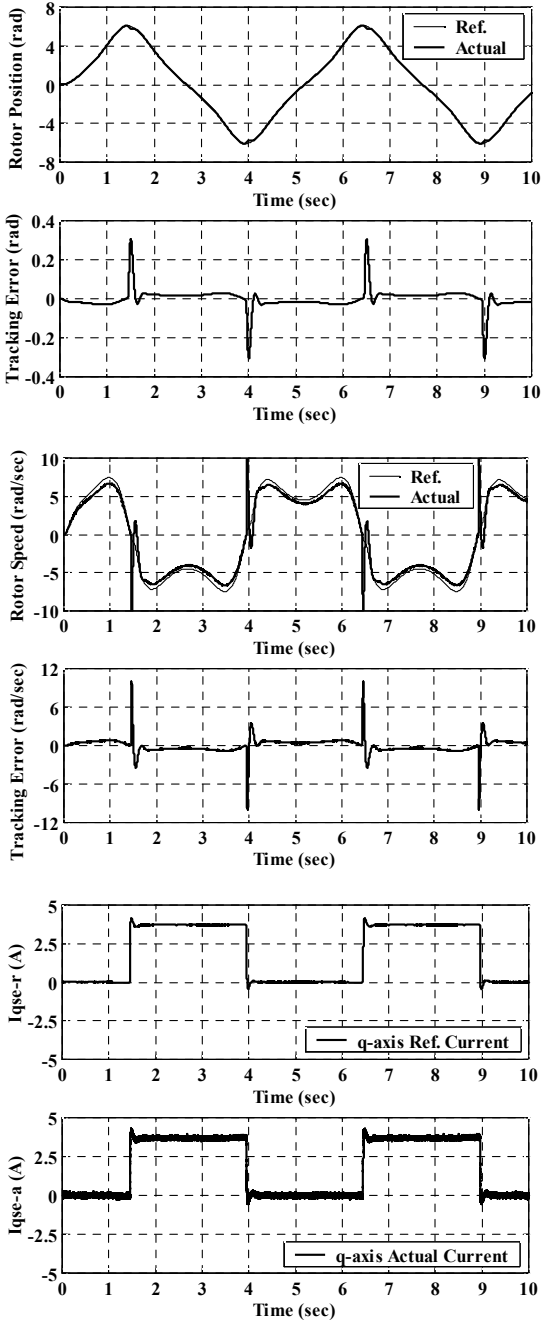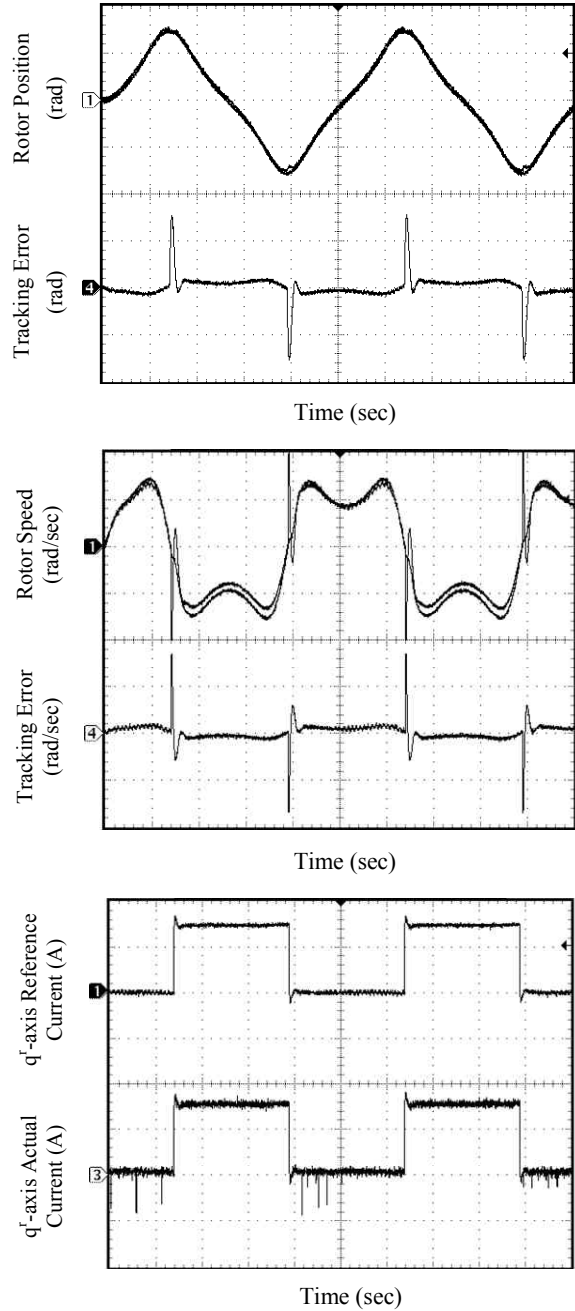
(a) Simulation Results.　　　　(b) Experimental Results.

Fig. 6. Dynamic response for the reference position of 2π rad and subsequent loading of 3.6 N.m for PMSM servo drive system at Case (1) of parameter uncertainties using L_2 compensated position controller.

Experimental Scales: position response 4 rad/div, speed response 5 (rad/sec)/div, tracking position error 0.2 rad/div, tracking speed error 6 (rad/sec)/div, adaptive position signal 3 rad/div, adaptive speed signal 1 (rad/sec)/div, q-d axis current response 2.5 A/div, time base for all traces 1 sec/div.
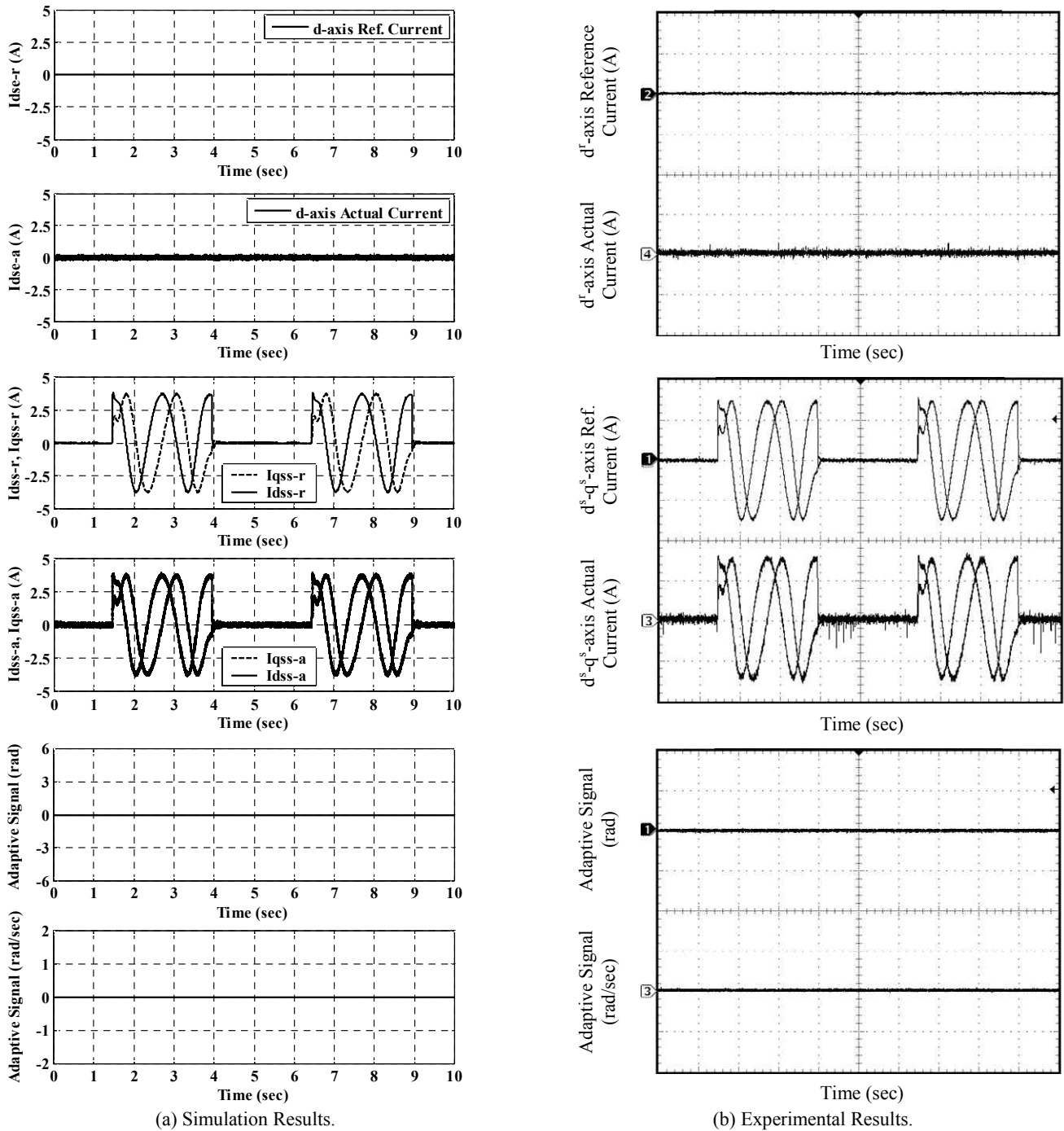
(a) Simulation Results.   (b) Experimental Results.

Fig. 6.   (*Continued*) Dynamic response for the reference position of 2π rad and subsequent loading of 3.6 N.m for PMSM servo drive system at Case (1) of parameter uncertainties using L_2 compensated position controller.
Experimental Scales: position response 4 rad/div, speed response 5 (rad/sec)/div, tracking position error 0.2 rad/div, tracking speed error 6 (rad/sec)/div, adaptive position signal 3 rad/div, adaptive speed signal 1 (rad/sec)/div, q-d axis current response 2.5 A/div, time base for all traces 1 sec/div.

The learning rate parameters are chosen as: $\eta_{\varpi} = 0.09$, $\eta_W = 0.05$, $\eta_\Theta = 0.1$, $\eta_\mu = 0.21$, $\eta_\sigma = 0.19$, $\sigma = 0.75$, $D = 0.075$, $\eta_1 = 0.01$, $\eta_2 = 0.025$, $\eta_3 = 0.035$, $\eta_4 = 0.08$, and $\eta_5 = 0.04$.

A. Simulation of the PMSM Servo Drive System

The simulation results of the PMSM drive system are presented to verify the feasibility of the proposed IRCS under various operating conditions. To investigate the robustness of the proposed controllers, four cases including PU and external load disturbances are considered.
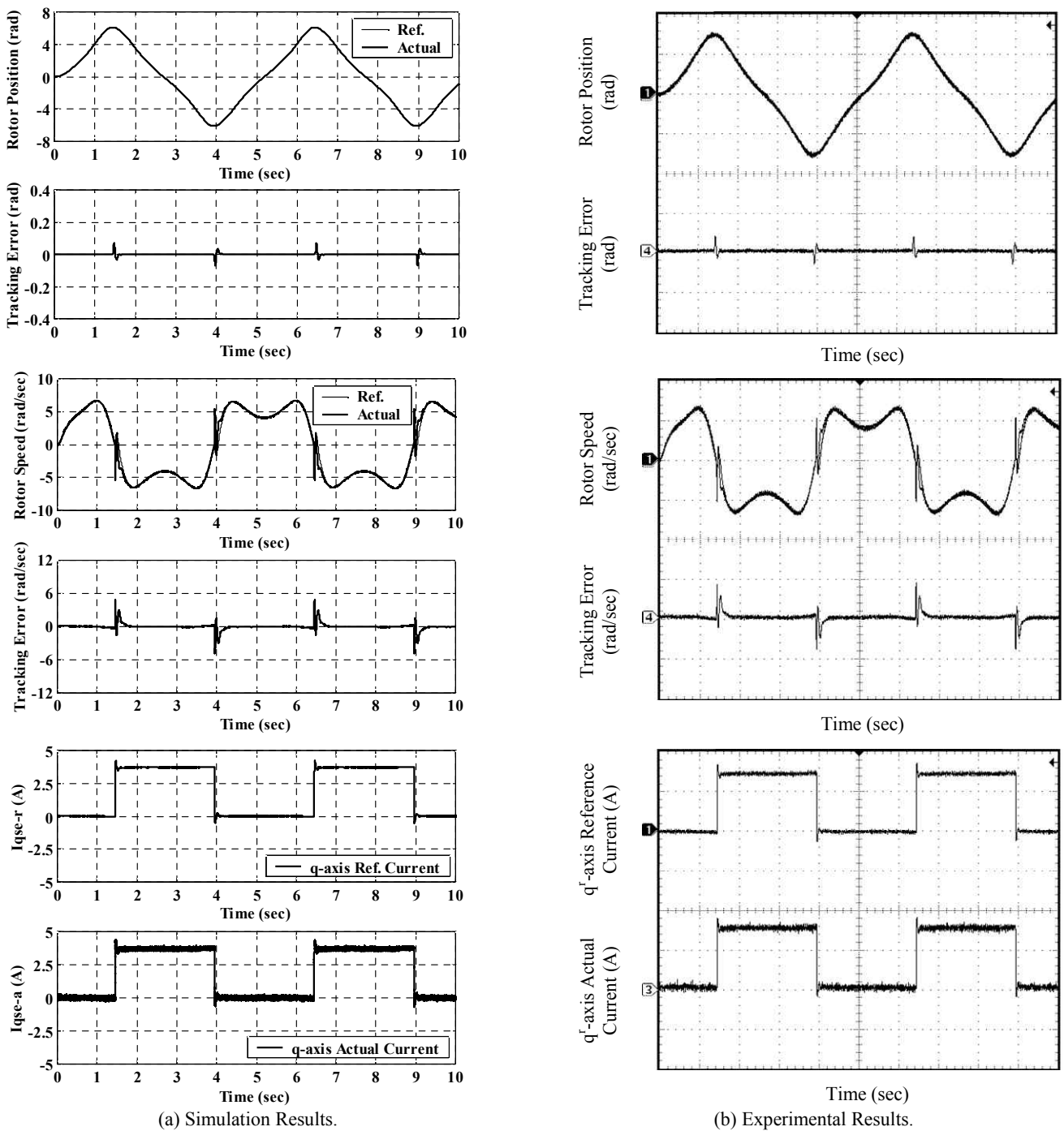
(a) Simulation Results.　　(b) Experimental Results.

Fig. 7.　Dynamic response for the reference position of 2π rad and subsequent loading of 3.6 N.m for PMSM servo drive system at Case (1) of parameter uncertainties using IRCS based on RWIT2FNN position tracking controller.

Experimental Scales: position response 4 rad/div, speed response 5 (rad/sec)/div, tracking position error 0.2 rad/div, tracking speed error 6 (rad/sec)/div, adaptive position signal 3 rad/div, adaptive speed signal 1 (rad/sec)/div, q-d axis current response 2.5 A/div, time base for all traces 1 sec/div
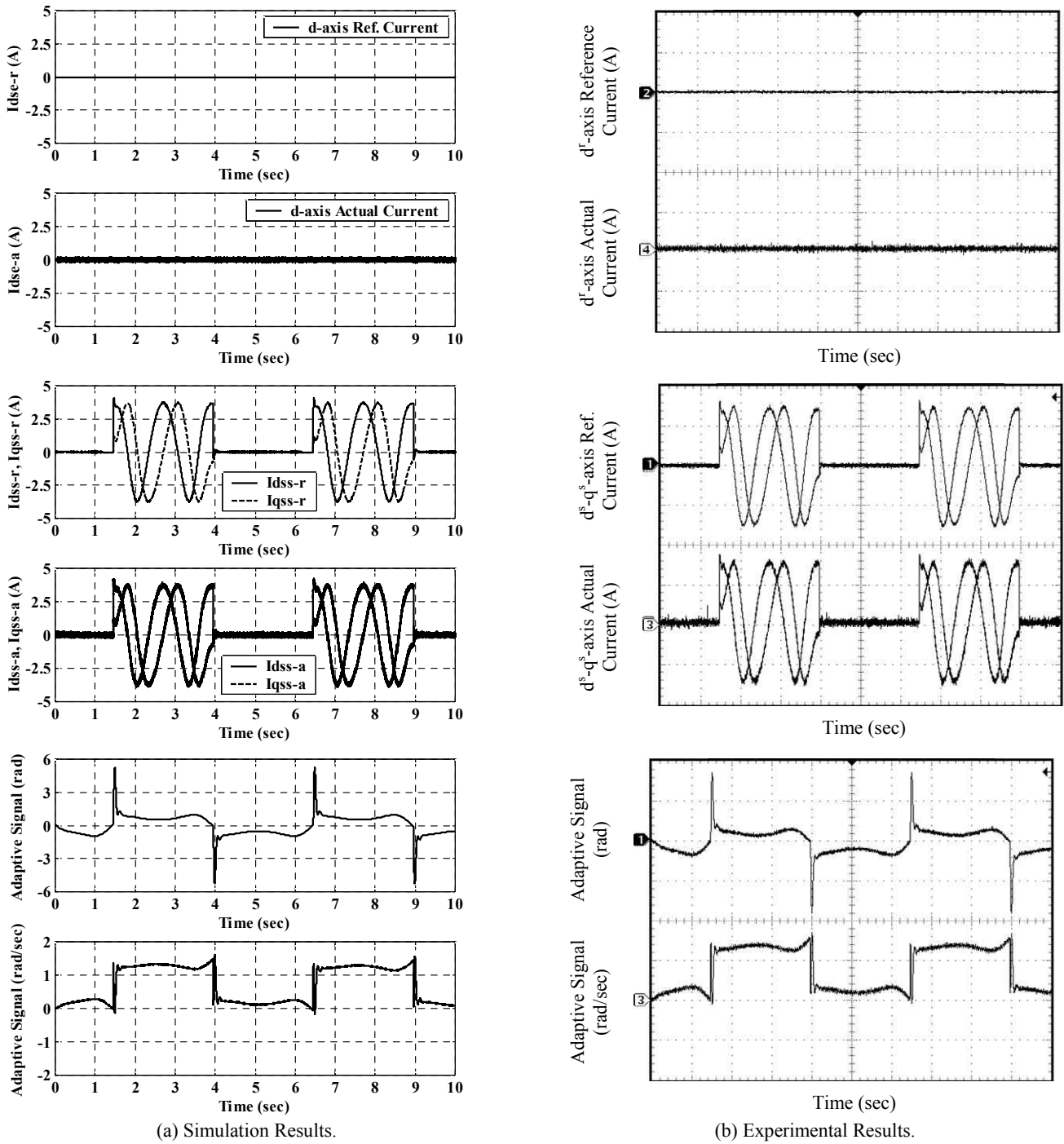
(a) Simulation Results.

(b) Experimental Results.

Fig. 7. (*Continued*) Dynamic response for the reference position of 2π rad and subsequent loading of 3.6 N.m for PMSM servo drive system at Case (1) of parameter uncertainties using IRCS based on RWIT2FNN position tracking controller.
Experimental Scales: position response 4 rad/div, speed response 5 (rad/sec)/div, tracking position error 0.2 rad/div, tracking speed error 6 (rad/sec)/div, adaptive position signal 3 rad/div, adaptive speed signal 1 (rad/sec)/div, q-d axis current response 2.5 A/div, time base for all traces 1 sec/div

Case 1: $1.0 \times (L_s/R_s)$, $1.0 \times (\beta_m/J_m)$, $1.00 \times \lambda_m$, T_L=0–3.6 N.m
Case 2: $0.5 \times (L_s/R_s)$, $1.0 \times (\beta_m/J_m)$, $0.85 \times \lambda_m$, T_L=0–3.6 N.m
Case 3: $1.5 \times (L_s/R_s)$, $1.0 \times (\beta_m/J_m)$, $1.25 \times \lambda_m$, T_L=0–3.6 N.m
Case 4: $1.5 \times (L_s/R_s)$, $1.0 \times (\beta_m/J_m)$, $1.25 \times \lambda_m$, T_L=0–3.6 N.m
The dynamic performance of the PMSM servo drive due to

a reference model command of 2π rad under a subsequent loading of 3.6 N.m for the compensated L_2 controller alone in Case (1) of the PU including the responses of the reference model and the rotor position, the tracking position error, the rotor speed, the tracking speed error, the d-q axis current
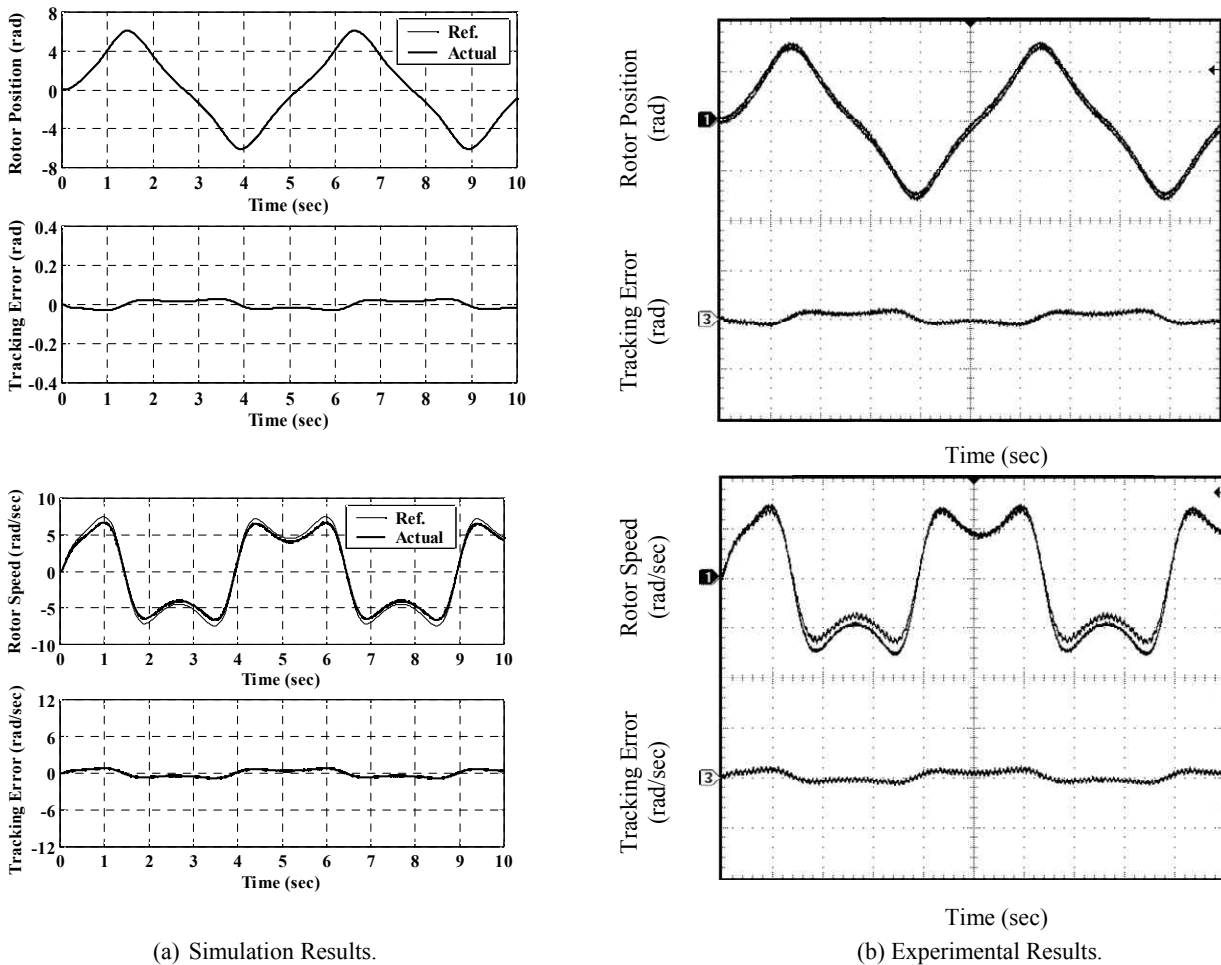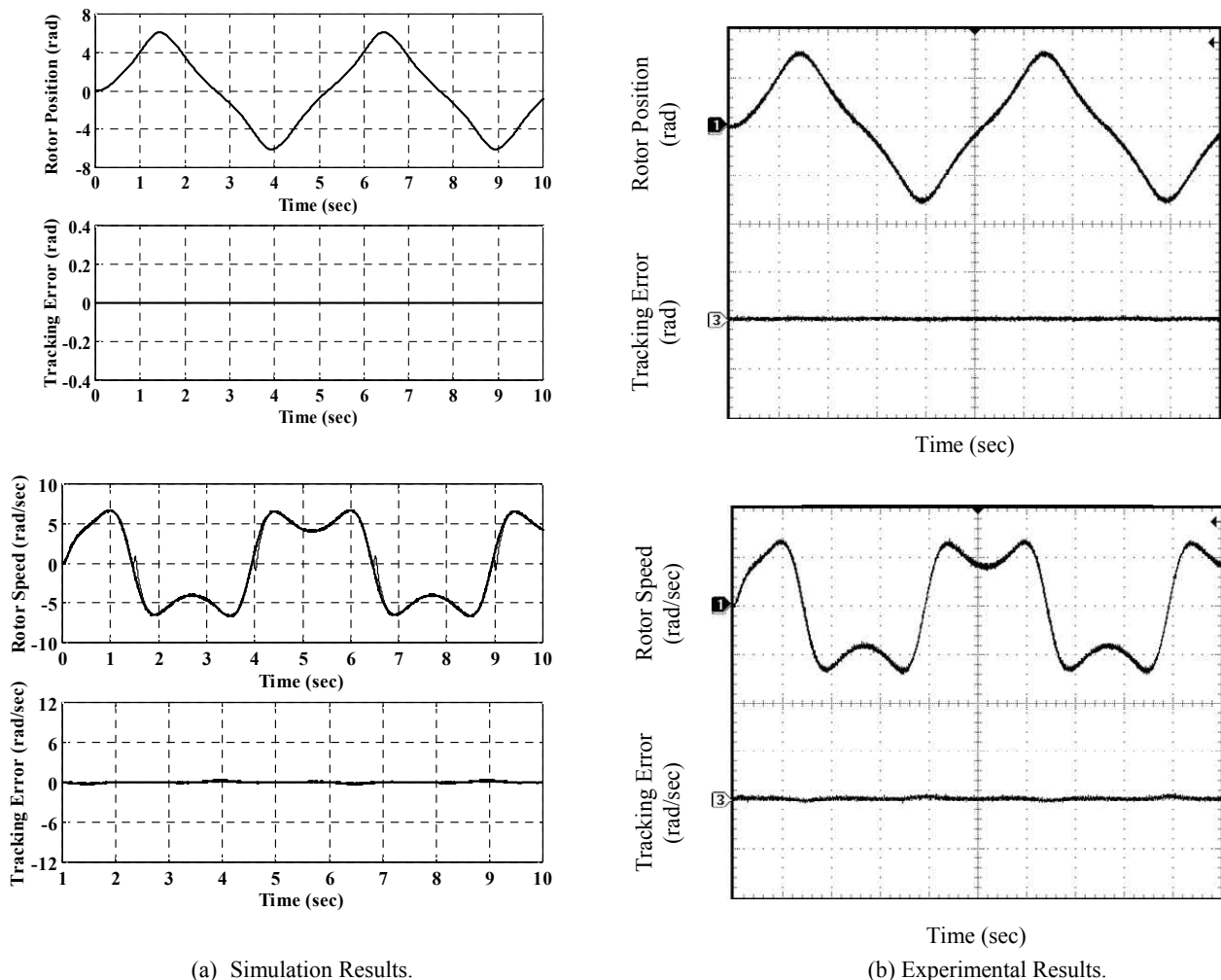
(a) Simulation Results.   (b) Experimental Results.

Fig. 8. Dynamic response for the reference position of 2π rad and no-loading for PMSM servo drive system at Case (1) of parameter uncertainties using L_2 compensated position controller.

Experimental Scales: position response 4 rad/div, speed response 5 (rad/sec)/div, tracking position error 0.2 rad/div, tracking speed error 6 (rad/sec)/div, adaptive position signal 3 rad/div, adaptive speed signal 1 (rad/sec)/div, q-d axis current response 2.5 A/div, time base for all traces 1 sec/div.

response and the adaptive control signals are predicted, as shown in Fig. 6(a). On the other hand, the dynamic performance of the PMSM servo drive using the IRCS is shown in Fig. 7(a) in Case (1) of the PU. The disturbance rejection capabilities have been checked when a load of 3.6 N.m is applied to the shaft at $t = 1.45$ sec. The results obtained in Figs. 6(a) and 7(a) illustrate good dynamic performances in the command tracking and load regulation. They are realized for both of the position tracking controllers. Improvement of the control performance by the addition of the proposed RWIT2FNNC can be observed from the obtained results in the command tracking and the load regulation characteristics. From the results shown in Fig. 7(a), the tracking position and speed errors with the compensated controller are larger than the ones obtained when using the RWIT2FNNC. The dynamic performance of the PMSM servo drive due to a reference model command of 2π rad under no-loading for the compensated L_2 controller alone in

Case (1) of the PU including the responses of the reference model and the rotor position, the tracking position error, the rotor speed and the tracking speed error are predicted, as shown in Fig. 8(a). On the other hand, the dynamic performance of the PMSM servo drive under the same operating conditions using the IRCS is shown in Fig. 9(a) in Case (1) of the PU.

To further verify the performance robustness of the proposed control schemes, four cases of PU and external load disturbances are considered, cases (1~4), for comparison. The dynamic performance of the PMSM servo drive for both of the position controllers at all Cases of PU is predicted in Fig. 10. Furthermore, the maximum tracking position errors under the four cases of PU are approximately 0.35 rad for the proposed L_2 compensated control system. On the other hand, the ones with the IRCS under the four examined cases of PU are approximately constant and equal 0.12 rad. The maximum position regulation dips under the four cases of PU are 0.25

<center>(a) Simulation Results.   (b) Experimental Results.</center>

Fig. 9. Dynamic response for the reference position of 2π rad and no-loading for PMSM servo drive system at Case (1) of parameter uncertainties using IRCS based on RWIT2FNN position tracking controller.
Experimental Scales: position response 4 rad/div, speed response 5 (rad/sec)/div, tracking position error 0.2 rad/div, tracking speed error 6 (rad/sec)/div, adaptive position signal 3 rad/div, adaptive speed signal 1 (rad/sec)/div, q-d axis current response 2.5 A/div, time base for all traces 1 sec/div

rad, 0.23 rad, 0.3 rad, and 0.35 rad, respectively for the proposed compensated control system. On the other hand, the ones with the IRCS under the four cases of PU are approximately constant and equal 0.12 rad. From the simulation results shown in Fig. 10, the tracking errors converges quickly and the robust control characteristics of the proposed IRCS under the occurrence of PU can be clearly observed. Compared with the compensated control system, the tracking errors and regulation characteristics are greatly reduced. Therefore, the proposed IRCS can yield control performance that is superior to that of the compensated control scheme. As a result, the proposed IRCS provides a rapid and accurate response for the reference model under load changes. It is within 0.5 sec which is quite fast when compared with the compensated controller which has a sluggish recovery time of more than 1.0 sec at PU. Thus it can be verified that the proposed IRCS under all cases of PU

can satisfy the robustness and accuracy requirements and is more suitable in the tracking control of the PMSM servo drives for industrial applications.

B. Experimentation on the PMSM Servo Drive System

To further verify the performance of the proposed control scheme applied to a PMSM servo drive in practical applications, some experimental results are introduced. The experimental results on the dynamic performance of the proposed compensated controller due to a reference model command under a subsequent loading of 3.6 N.m at Case (1) of the PU including the responses of the reference model and the rotor position, the tracking position error, the rotor speed, the tracking speed error, the d-q axis current response and the adaptive signals are predicted in Fig. 6(b). On the other hand, the experimental results on a PMSM servo drive using the proposed IRCS are shown in Fig. 7(b) under the same

conditions. Furthermore, the disturbance rejection capabilities have been checked for both of the position controllers. In addition, the maximum tracking position errors at case (1) of the PU is approximately 0.36 rad, for the proposed compensated controller. On the other hand, the one with the IRCS at case (1) of the PU is approximately 0.13 rad. The maximum position regulation dip under the same condition is 0.37 rad for the proposed compensated controller. On the other hand, the one with the IRCS is approximately 0.13 rad. The experimental results obtained in Figs. 6(b) and 7(b) clearly illustrate the good dynamic performances, in terms of command tracking and load regulation performance, are realized for both position tracking controllers. When compared with the compensated controller, the tracking errors and regulation characteristics are much reduced for the proposed IRCS. Therefore, the IRCS can yield superior control performance than the L_2 compensated controller. The dynamic performance of the PMSM servo drive due to a reference model command of 2π rad under no-loading for the compensated L_2 controller alone at Case (1) of the PU including the responses of the reference model and the rotor position, the tracking position error, the rotor speed and the tracking speed error are predicted, as shown in Fig. 8(b). On the other hand, the dynamic performance of the PMSM servo drive under the same operating conditions using the IRCS is shown in Fig. 9(b) at Case (1) of the PU. As a result, the proposed IRCS provides a rapid and accurate response for the reference model under load changes. It is within 0.5 sec which is quite fast when compared with the compensated position tracking controller which has a sluggish recovery time of more than 1.0 sec. It is obvious that the performance of the PMSM servo drove system using the compensated controller is greatly improved by using the IRCS. Thus it can be verified that the proposed IRCS can satisfy the accuracy requirements and is more suitable in the tracking control of PMSM servo drive systems in practical applications.

C. Performance Measure of the PMSM Servo Drive System

To measure the performance of the PMSM servo drive, the maximum tracking error, TE_{max}, the average tracking error, TE_{mean}, and the standard deviation of the tracking error, TE_{sd}, are defined as follows:

$$TE_{\max} = \max_k \sqrt{T(k)^2} \tag{89}$$

$$TE_{mean} = \sum_{k=1}^{n} \frac{T(k)}{n} \tag{90}$$

$$TE_{sd} = \sqrt{\sum_{k=1}^{n} \frac{(T(k) - TE_{mean})^2}{n}} \tag{91}$$

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

TABLE II
PERFORMANCE MEASURES OF THE PMSM SERVO DRIVE SYSTEM AT CASE (1) OF PU

Controller Type	Tracking Errors (rad)		
	Maximum	Average	S.D.
2DOF I-PDC	0.62680	0.0032250	0.332500
SMC	0.49740	0.0004173	0.115900
L_2 Controller	0.30540	2.290e-05	0.052480
CTC	0.21030	0.0002138	0.032390
ENNC	0.19230	0.0009900	0.027930
FNNC	0.15990	0.0019580	0.075830
RWENNC	0.10260	0.0003200	0.010440
IRCS	0.06844	2.0627e-05	0.007954

To further investigate the improvement of the proposed IRCS, performance measures of the Elman neural network controller (ENNC), the recurrent wavelet ENNC (RWENNC), the computed torque controller (CTC), the conventional two-degrees-of-freedom integral plus proportional and rate feedback controller (2DOF I-PDC), the L_2 compensated controller, the fuzzy neural network controller (FNNC) and the sliding-mode controller (SMC) are compared and summarized in Table II. From the results shown in Table II, it can be easily seen that high values of TE_{max}, TE_{mean} and TE_{sd} have been successfully reduced by the proposed IRCS. Therefore, the IRCS possesses the best robust control characteristics and can control PMSM servo drive systems effectively.

V. CONCLUSIONS

This paper proposed an IRCS for PMSM servo drives which guarantees robustness in the presence of parameter uncertainties and load disturbances. The proposed control scheme comprises an RWIT2FNNC, an RWIT2FNNE and a compensated controller. The RWIT2FNNC combines the merits of a self-constructing interval type-2 fuzzy logic system, a recurrent neural network and a WNN. Moreover, it performs the structure and parameter-learning concurrently. The RWIT2FNNC is used as the main position tracking controller to mimic the ICL. In addition, the RWIT2FNNE is developed to approximate an unknown dynamic function including parameter uncertainty. Furthermore, an L_2 compensated controller is designed to achieve L_2 tracking performance with a desired attenuation level. Moreover, the adaptive learning algorithms for the L_2 compensated controller and the RWIT2FNNE are derived based on the Lyapunov stability analysis so that the stability of the PMSM servo drive can be guaranteed. Simulation and experimental results confirm that the proposed IRCS grants robust performance and a precise dynamic response to the reference model regardless of load disturbances and PMSM parameter uncertainties.
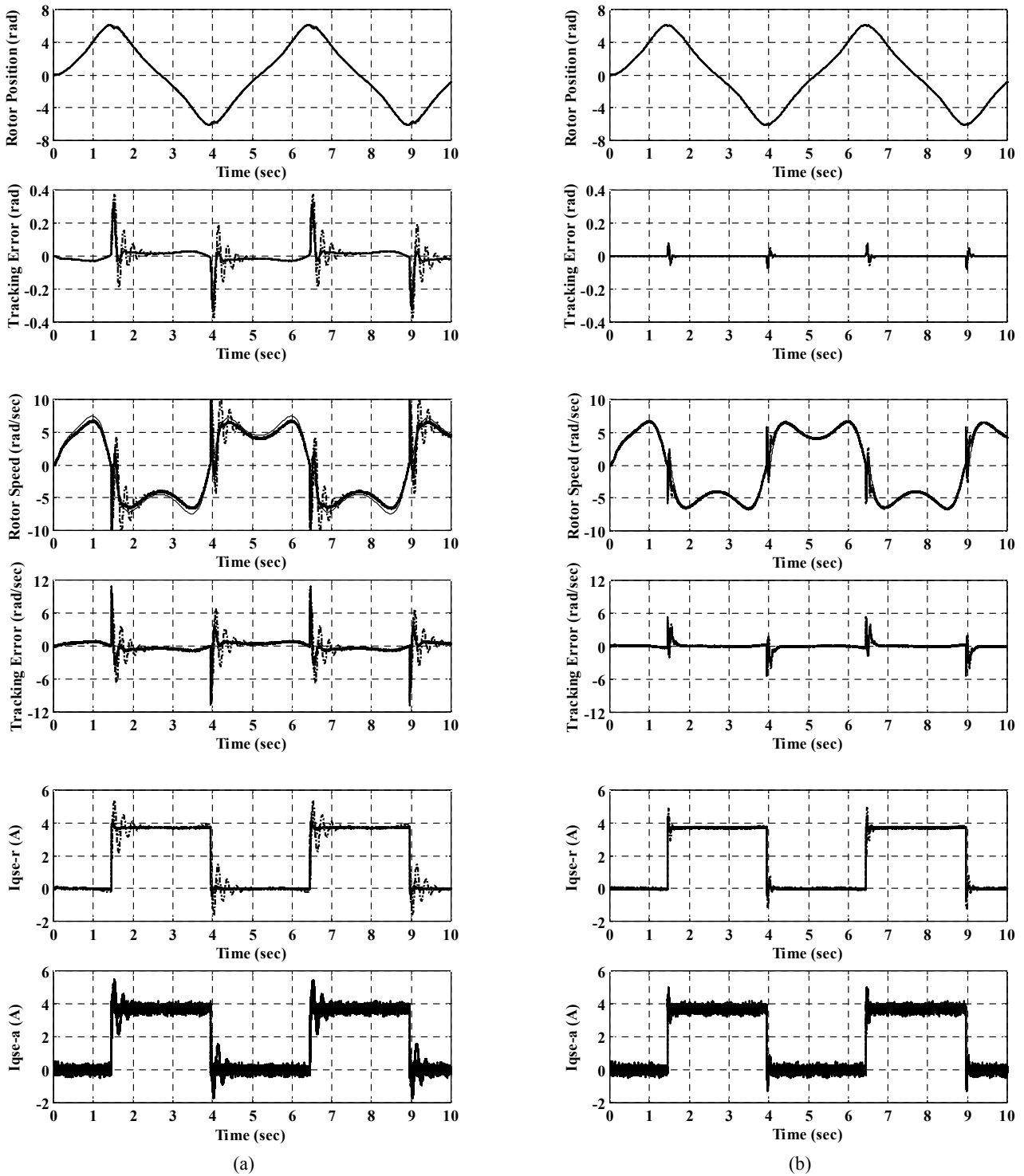
Fig. 10. Dynamic response for the reference position of 2π rad and subsequent loading of 3.6 N.m for both position controllers at different Cases (1~4) of parameter uncertainties.
(a) Using L_2 compensated position controller
(b) Using IRCS based on RWIT2FNN position tracking controller

REFERENCES

[1] W. Leonhard, *Control of Electrical Drives*, Springer-Verlag, Berlin, 1996.

[2] R. Krishnan, *Electric Motor Drives: Modeling, Analysis, and Control*, Prentice-Hall, New Jersey, 2001.

[3] F. F. M. El-Sousy, "Robust wavelet-neural-network sliding-mode control system for permanent-magnet synchronous motor drive," *IET—Electric Power Application*, Vol. 5, No. 1, pp. 113-132, Jan. 2011.

[4] K. Jezernik and M. Rodic, "High precision motion control of servo drives," *IEEE Trans. Ind. Electron.*, Vol. 56, No. 10, pp. 3810-3816, Oct. 2009.

[5] J. W. Finch and D. Giaouris, "Controlled AC Electrical Drives," *IEEE Trans. Ind. Electron.*, Vol. 55, No. 2, pp. 481-491, 2008.

[6] X. Lin-Shi, F. Morel, A. M. Llor, B. Allard, J.-M. Retif, "Implementation of hybrid control for motor drives," *IEEE Trans. Ind. Electron.*, Vol. 54, No. 4, pp. 1946-1952, Aug. 2007.

[7] M. Teshnehllab and K. Watanabe, "Self tuning of computed torque gains by using neural networks with flexible structures," *Proc. Inst. Elect. Eng. Control Theory Appl.*, Vol. 141, No. 4, pp. 235-242, 1994.

[8] F. J. Lin, Y. S. Lin, and S. L. Chiu, "Slider-crank mechanism control using adaptive computed torque technique," *Proc. Inst. Elect. Eng. Control Theory Appl.*, Vol. 145, No. 3, pp. 364-376, 1998.

[9] R. J. Wai, "Hybrid control for speed sensorless induction motor drive," *IEEE Trans. Fuzzy Syst.*, Vol. 9, No. 1, pp. 116-138, Feb. 2001.

[10] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, Vol. 1, No. 1, pp. 4-27, 1990.

[11] F. F. M. El-Sousy, "Robust adaptive wavelet-neural-network sliding-mode control for a dsp-based pmsm drive system," *Journal of Power Electronics*, Vol. 10, No. 5, pp. 518-527, Sep. 2010.

[12] F. F. M. El-Sousy, "A vector-controlled pmsm drive with a continually on-line learning hybrid neural-network model-following speed controller," *Journal of Power Electronics*, Vol. 5, No. 2, pp. 197-210, Apr. 2005.

[13] F. F. M. El-Sousy, "Robust tracking control based on intelligent sliding-mode model-following position controller for pmsm servo drives," *Journal of Power Electronics*, Vol. 7, No. 2, pp. 159-173, Apr. 2007.

[14] K. T. Tanaka and H. O. Wang, *Fuzzy Control Systems Design and Analysis*, New York: Wiley, 2001.

[15] Y. S. Lu and J. S. Chen, "A self-organizing fuzzy sliding-mode controller design for a class of nonlinear servo systems," *IEEE Trans. Ind. Electron.*, Vol. 41, No. 5, pp. 492-502, Oct. 1994.

[16] F. J. Lin, W. J. Hwang, and R. J. Wai, "A supervisory fuzzy neural network control system for tracking periodic inputs," *IEEE Trans. Fuzzy Syst.*, Vol. 7, No. 1, pp. 41-52, Feb. 1999.

[17] 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, Feb. 2001.

[18] R. J. Wai and F. J. Lin, "Fuzzy neural network sliding mode position controller for induction servo motor drive," *Proc. Inst. Elect. Eng. Electr. Power Applicat.*, Vol. 146, No. 3, pp. 297-308, 1999.

[19] 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, Feb. 2002.

[20] 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, Oct. 1999.

[21] J. Zhang and A. J. Morris, "Recurrent neuro-fuzzy networks for nonlinear process modeling," *IEEE Trans. Neural Netw.*, Vol. 10, No. 2, pp. 313-326, Mar. 1999.

[22] F. J. Lin and C. H. Lin, "Online gain-tuning IP controller using RFNN," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. 37, No. 2, pp. 655-670, Apr. 2001.

[23] C. H. Lee and C.C. Teng, "Identification and control of dynamic systems using recurrent-fuzzy-neural -network," *IEEE Trans. Fuzzy Syst.*, Vol. 8, No. 4, pp. 349-366, Aug. 2000.

[24] F. J. Lin, P. K., Huang and W. D. Chou, "Recurrent-Fuzzy-Neural-Network-Controlled Linear Induction Motor Servo Drive Using Genetic Algorithms," *IEEE Trans. Ind. Electron.*, Vol. 54, No. 3, pp. 1449-1461, Jun. 2007.

[25] 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.

[26] C. J Lin. and Y. C. Hsu, "Reinforcement hybrid evolutionary learning for recurrent wavelet-based neurofuzzy systems," *IEEE Trans. Fuzzy Syst.*, Vol. 15, No. 4, pp. 729-745, Aug. 2007.

[27] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Inf. Sci.*, Vol. 8, No. 3, pp. 199-249, 1975.

[28] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *IEEE Trans. Fuzzy Syst.*, Vol. 10, No. 2, pp. 117-127, Apr. 2002.

[29] J. M. Mendel, R. I. B. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Trans. Fuzzy Syst.*, Vol. 14, No. 6, pp. 808-821, Dec. 2006.

[30] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Englewood Cliffs, NJ: Prentice-Hall, 2001.

[31] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Inf. Sci.*, Vol. 132, No. 1, pp. 195-220, 2001.

[32] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, Vol. 8, No. 5, pp. 535-550, Oct. 2000.

[33] M. Singh, S. Srivastava, J. R. P. Gupta, and M. Hanmandlu, "A type-2 fuzzy neural model based control of a nonlinear system," *IEEE Int. Conf. Cybern. Intell. Syst.*, Vol. 2, pp. 1352-1356, 2004.

[34] M. Karakose and E. Akin, "Type-2 fuzzy activation function for multilayer feedforward neural networks," *IEEE Int. Conf. Syst., Man Cybern.*, Vol. 4, pp. 3762-3767, Oct. 10-13, 2004.

[35] C. Lynch, H. Hagras, and V. Callaghan, "Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines," *IEEE Int. Conf. Fuzzy Syst.*, pp. 1446-1453, Jul. 2006.

[36] C. H. Wang, C. S. Cheng, and T. T. Lee, "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*,

Vol. 34, No. 3, pp. 1462-1477, Jun. 2004.

[37] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure & Appl. Math.*, Vol. 41, No. 7, pp. 909-996, 1988.

[38] T. Yamakawa, E. Uchino, and T. Samatsu, "Wavelet neural networks employing over-complete number of compactly supported nonorthogonal wavelets and their applications," *IEEE Int. Conf. Neural Networks*, Vol. 3, pp. 1391-1396, 1994.

[39] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Syst.*, Vol. 6, No. 1, pp. 12-32, Feb. 1998.

[40] V. V. Cross and T. A. Sudkamp, *Similarity and Compatibility in Fuzzy Set Theory: Assessment and Application*, Heidelberg, Germany: Physica-Verlag, 2002.

[41] D. Wu and J. M. Mendel, "A vector similarity measure for interval type-2 fuzzy sets," *IEEE Int. Conf. Fuzzy Syst.*, pp. 1-6, Jul. 2007.

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

[43] K. J. Astrom and B. Wittenmark, *Adaptive Control*, New York: Addison Wesley, 1995.

[44] B. S. Chen and C. H. Lee, and Y.-C. Chang, "H^∞ tracking design of uncertain nonlinear SISO systems: Adaptive fuzzy approach," *IEEE Trans. Fuzzy Syst.*, Vol. 4, No. 1, pp. 32-43, Feb. 1996.

[45] A. Rubaai, "Direct adaptive fuzzy control design achieving H^∞ tracking for high performance servo drives," *IEEE Trans. Energy Convers.*, Vol. 14, No. 4, pp. 1199-1208, Dec. 1999.

[46] C. Chang and B. S. Chen, "A nonlinear adaptive H_∞ tracking control design in robotic systems via neural networks," *IEEE Trans. Contr. Syst. Technol.*, Vol. 5, No. 1, pp. 23-29, 1997.

[47] M. C. Hwang, X. Hu, and Y. Shrivastava, "Adaptive H_∞ neural network tracking controller for electrically driven manipulators," *Proc. IEE—Control Theory Application*, Vol. 145, No. 6, pp. 594-602, 1998.

[48] J. A. Ball, P. Kachroo, and A. J. Krener, "H_∞ tracking control for a class of nonlinear systems," *IEEE Trans. Automatic Control*, Vol. 44, No. 6, pp. 1202-1206, Jun. 1999.

[49] F. F. M. El-Sousy, "Hybrid recurrent cerebellar model articulation controller-based supervisory H^∞ motion control system for permanent-magnet synchronous motor servo drive," *IET—Electric Power Application,* Vol. 5, No. 7, pp. 563-579, Aug. 2011.

[50] F. F. M. El-Sousy, "Robust adaptive H^∞ position control via a wavelet-neural-network for a DSP-based permanent-magnet synchronous motor servo drive system," *IET-Electric Power Application*, Vol. 4, No. 5, pp. 333-347, May 2010.

[51] F. F. M. El-Sousy, "Hybrid H^∞-based wavelet-neural-network tracking control for permanent-magnet synchronous motor drives," *IEEE Trans. Ind. Electron.*, Vol. 57, No. 9, pp. 3157-3166, Sep. 2010.

[52] F. F. M. El-Sousy, "Intelligent model-following position control for pmsm servo drives," in *Proc. 6th WSEAS Int. Conf. Neural Netw.*, pp. 230-238, Jul. 2005.

[53] F. F. M. El-Sousy, "An intelligent model-following sliding-mode position controller for pmsm servo drives," *4th IEEE International Conference on Mechatronics*, May 2007.

Fayez F. M. El-Sousy was born in Gahrbia Prefecture, Egypt, in 1965. He received his B.S. in Electrical Power and Machines Engineering from Menoufia University, Egypt, in 1988, and his M.S. and Ph.D. in Electrical Power and Machines Engineering from Cairo University, Egypt, in 1994 and 2000, respectively. Since 1990, he has been with the Department of Power Electronics and Energy Conversion at the Electronics Research Institute (ERI), Egypt, where he is currently an Associate Professor. From April 2004 to February 2007, he was a Post Doctoral Visiting Researcher at the Energy Conversion Laboratory, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan. Since September 2007 he has been with the department of Electrical Engineering, College of Engineering, Salman bin Abdulaziz University, Saudi Arabia. He is also the Chairman of the Department of Electrical Engineering. His research interests include the modeling and control of motor drives, motion-control systems, wind energy systems, DSP-based computer control systems, intelligent control theories including fuzzy logic, neural networks and wavelets, nonlinear control theories and power electronics. Dr. El-Sousy is currently interested in the intelligent control of Maglev vehicle transportation systems.