

Control of Doubly-Fed Induction Generator System Using PFNN

Faa-Jeng Lin[#] [△]

Kuang-Hsiung Tan^{*}

Zong-Han Lu[#] [◎]

Yung-Ruei Chang[§]

[#]Dept. of Electrical Engineering ^{*}Dept. of Electrical and Electronic Engineering [§]Engineering Technology and Facility Operation Division

National Central University
Chungli, Taiwan

National Defense University
Taoyuan, Taiwan

Institute of Nuclear Energy Research
Taoyuan, Taiwan

[△]linfj@ee.ncu.edu.tw

s913115@gmail.com.tw

raymond@iner.gov.tw

[◎]scott31025@hotmail.com

Abstract—An intelligent controlled doubly-fed induction generator (DFIG) system using probabilistic fuzzy neural network (PFNN) is proposed in this study. This system can be applied as a stand-alone power supply system or as the emergency power system when the electricity grid fails for all sub-synchronous, synchronous and super-synchronous conditions. The rotor side converter is controlled using the field-oriented control to produce three-phase stator voltages with constant magnitude and frequency at different rotor speeds. Moreover, the stator side converter, which is also controlled using field-oriented control, is primarily implemented to maintain the magnitude of the DC-link voltage. Furthermore, an intelligent PFNN controller is proposed for both the rotor and stator side converters to improve the transient and steady-state responses of the DFIG system for different operating conditions. The network structure, on-line learning algorithm and convergence analyses of the PFNN are introduced in detail. Finally, the feasibility of the proposed control scheme is verified using some experimental results.

Keywords—doubly-fed induction generator; field-oriented control; probabilistic fuzzy neural network

NOMENCLATURE

$v_{abs}, v_{bcs}, v_{cas}$	3-Phase stator line voltage.
i_{as}, i_{bs}, i_{cs}	3-Phase stator current.
i_{al}, i_{bl}, i_{cl}	3-Phase load current.
i_{ar}, i_{br}, i_{cr}	3-Phase rotor current.
i_a, i_b, i_c	3-Phase current of stator side converter.
L_a, L_b, L_c	3-Phase inductor at stator side converter.
v_{an}, v_{bn}, v_{cn}	3-Phase stator phase voltage.
C	DC-link capacitance.
V	Magnitude of stator phase voltage.
V_n	Magnitude of phase A stator voltage.
V_n^*	Command magnitude of phase A stator voltage.
V_{dc}	DC-link voltage.
V_{dc}^*	Command of DC-link voltage.
d, q	d - q Axis.

v_{ds}, v_{qs}	2-Axis stator voltage.
R_s	Stator resistance.
$\lambda_{ds}, \lambda_{qs}$	2-Axis stator flux linkage.
i_{ds}, i_{qs}	2-Axis stator current.
i_{dr}, i_{qr}	2-Axis rotor current.
L_s	Stator inductance.
L_m	Magnetizing inductance.
P	Pole numbers.
θ_e, ω_e	Electric angle and angular speed of stator phase voltage.
θ_r, ω_r	Rotor angle and angular speed.
θ_{sl}, ω_{sl}	Slip angle and angular speed.
θ_m, ω_m	Mechanical angle and angular speed.
P_s	Real power of stator side converter.
Q_s	Reactive power of stator side converter.
v_d, v_q	2-Axis voltage of stator side converter.
i_d, i_d^*	d -Axis current and command current of stator side converter.
i_q, i_q^*	q -Axis current and command current of stator side converter.

I. INTRODUCTION

Recently, the concept of incorporating fuzzy logic into a neural network (NN) has been grown into a popular research topic. In contrast to the pure NN or fuzzy system, the fuzzy neural network (FNN) possesses both advantages including the capability of fuzzy reasoning in handling uncertain information and the capability of artificial neural networks in learning from processes. Moreover, FNNs do not require mathematical models and have the ability to approximate nonlinear and uncertainties systems. Therefore, there were many researches which are using FNNs to represent complex plants and construct advanced controllers based on the back-propagation (BP) algorithm. Furthermore, probabilistic neural network (PNN) is a feedforward neural network, and its structure is neural network implementation of Parzen nonparametric

probability density function (PDF) estimation and Bayes classification rule [1]. The PNN can handle the uncertainties in industry applications effectively, and have been widely used in pattern classification, nonlinear mapping, and fault detection and classification. Due to the above advantages of FNN and PNN, the PFNN, which combines the characteristics of FNN (degree of truth) and PNN (probability of truth) [2], has been proposed in some applications, such as modeling and control problems [3] and stochastic modeling.

The DFIG systems have been widely used on large variable-speed fixed-frequency wind power generation systems, hydro power systems and turbine engine power generation systems. In contrast to the squirrel-cage induction generator (SCIG) system and the synchronous generator system, the power capacity of the rotor side and stator side converters of the DFIG system required only 30% of the rated power [4]. This means that the desired variable-speed range is $\pm 30\%$ around the rated turbine speed. Thus, compared to the SCIG system where the converters have to handle the total power, the costs and losses in the power electronics converters can be much reduced.

There are two types of applications of a DFIG system, grid-connected type and stand-alone type. Several field-oriented control schemes for the DFIG have been proposed for the applications of both types [3], and the control strategies of the stator side and rotor side converters for the grid-connected type and stand-alone type are quite different.

The grid-connected DFIG system is quit mature and is widely used today and a lot of control schemes have been reported. However, only little attention has been paid toward the development of the control schemes of the stand-alone type DFIG system. On the other hand, satisfied transient performance can't be achieved using the traditional PID control for the DFIG system due to time-varying and nonlinear control characteristics of the DFIG system. Therefore, this study addresses the issue to develop an intelligent control scheme for the DFIG system for stand-alone application.

II. FIELD-ORIENTED CONTROL DFIG SYSTEM

A direct drive DFIG system using both the rotor side and stator side converters is shown in Fig. 1 and is consists of a doubly-fed induction machine with slip rings, and a back-to-back PWM converter between the rotor slip rings and the transformer. The stator is directly connected to the 3-phase load.

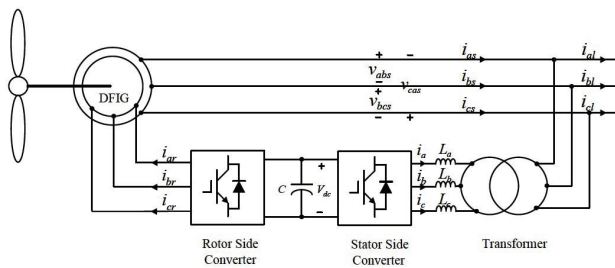


Figure 1. System block of DFIG.

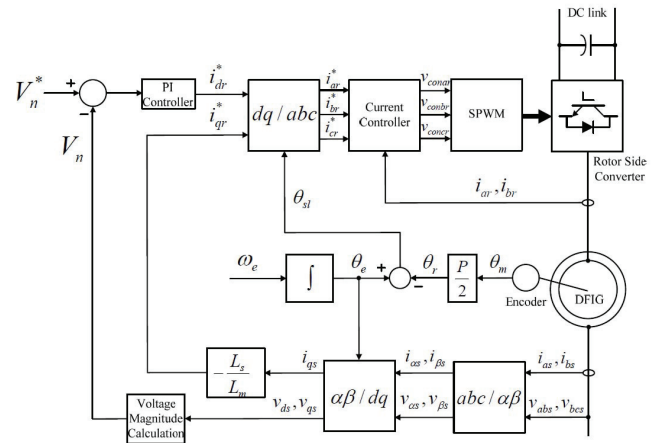


Figure 2. Control block of rotor side converter.

The control block diagram of the rotor side converter, which comprises a DFIG, a 3-phase current controlled voltage source inverter (VSI) using sinusoidal pulse-width-modulation (SPWM) and field-oriented control mechanism, is shown in Fig. 2 [5-6]. There are two control loops: one is for the regulation of the magnitude of the stator voltage; and the other one is for the field-oriented control using the command angular speed for the stator output voltage. The machine model of a DFIG can be described in the synchronous rotating reference frame as:

$$v_{ds} = R_s i_{ds} + \frac{d\lambda_{ds}}{dt} - \omega_e \lambda_{qs} \quad (1)$$

$$v_{qs} = R_s i_{qs} + \frac{d\lambda_{qs}}{dt} + \omega_e \lambda_{ds} \quad (2)$$

$$\lambda_{ds} = L_s i_{ds} + L_m i_{dr} \quad (3)$$

$$\lambda_{qs} = L_s i_{qs} + L_m i_{qr} \quad (4)$$

For the purpose of field-orientation in the synchronous rotating reference frame of the stator flux, the spatial vector of the stator flux is aligned with the d -axis in steady-state. That is

$$\lambda_{qs} = 0 \quad (5)$$

Substitute (5) into (4), then

$$i_{qr} = -\frac{L_s}{L_m} i_{qs} \quad (6)$$

It is possible to keep the spatial vector of the stator flux aligned to the reference frame d -axis despite of the load and rotor speed conditions if (6) is imposed. Moreover, the voltage control loop is used to control the magnitude of the stator voltage defined as follows:

$$V_n = \sqrt{(v_{ds})^2 + (v_{qs})^2} \quad (7)$$

Substituting (5) into (1), (2) and (3), the following equations are yielded

$$v_{ds} = R_s i_{ds} + \frac{d\lambda_{ds}}{dt} \quad (8)$$

$$v_{qs} = R_s i_{qs} + \omega_e \lambda_{ds} \quad (9)$$

$$\lambda_{ds} = L_s i_{ds} + L_m i_{dr} \quad (10)$$

Moreover, substitute (10) into (8) and (9), then

$$v_{ds} = R_s i_{ds} + L_s \frac{di_{ds}}{dt} + L_m \frac{di_{dr}}{dt} \quad (11)$$

$$v_{qs} = R_s i_{qs} + \omega_e L_s i_{ds} + \omega_e L_m i_{dr} \quad (12)$$

It can be seen from (11) and (12) that the spatial components of the stator voltage v_{ds} and v_{qs} are dependent on i_{dr} , i_{ds} and i_{qs} . However, i_{ds} and i_{qs} are load dependent perturbations. Therefore, only i_{dr} can be adopted as an independent control variable as shown in Fig. 2. Furthermore, for the sub-synchronous condition, the slip angular speed ω_{sl} is positive and the slip angular angle θ_{sl} is increasing, the direction of the power flow is from the DC-link to the rotor. On the other hand, for the super-synchronous conditions, the slip angular speed ω_{sl} is negative and the slip angular angle θ_{sl} is decreasing, the direction of the power flow is from the rotor to the DC-link.

The block diagram of the field-oriented control stator side converter with the stator voltage orientation is shown in Fig. 3. The 3-phase current controlled VSI using SPWM is also adopted in the current loop. Moreover, a voltage loop is designed for the regulation of the DC-link voltage. The active and reactive powers of the stator side converter can be described as [4,7]

$$P_s = \frac{3}{2} (v_d i_d + v_q i_q) \quad (13)$$

$$Q_s = \frac{3}{2} (v_d i_q - v_q i_d) \quad (14)$$

The d -axis voltage component v_d is fixed with the orientation of the stator voltage and the q -axis v_q voltage component is zero. Therefore, (13) and (14) can be simplified as follows:

$$P_s = \frac{3}{2} v_d i_d \quad (15)$$

$$Q_s = \frac{3}{2} v_d i_q \quad (16)$$

From (15) and (16), if i_q is set to zero, then the reactive power Q_s is zero. Thus, unit power factor of the stator side converter can be achieved. Moreover, if i_d is positive, the direction of the power flow is from the DC-link to the stator side. On the other hand, if i_d is negative, the direction of the power flow is from the stator side to the DC-link.

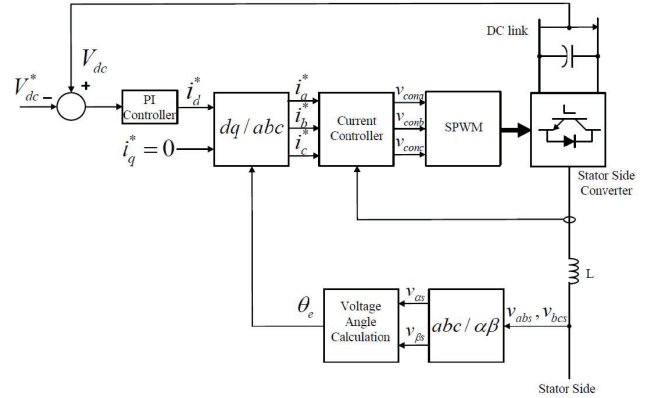


Figure 3. Control block of stator side converter.

III. PFNN CONTROLLER

In this section, the PFNN controller is proposed to replace the PI controllers shown in Figs. 2 and 3 for both the rotor side and stator side converters in order to improve the transient and steady-state responses of the DFIG system. In the proposed PFNN, not only the tracking error but also its derivative is considered as the inputs to achieve precise tracking performance in both the transient and steady-state phases to deal with instant uncertainties including system parameters variations and external disturbances. Moreover, in order to make the output of the PFNN sensitive to the history data, self-connection of the output neuron is also added. Therefore, the proposed PFNN can improve the convergence error and reduce the learning time effectively.

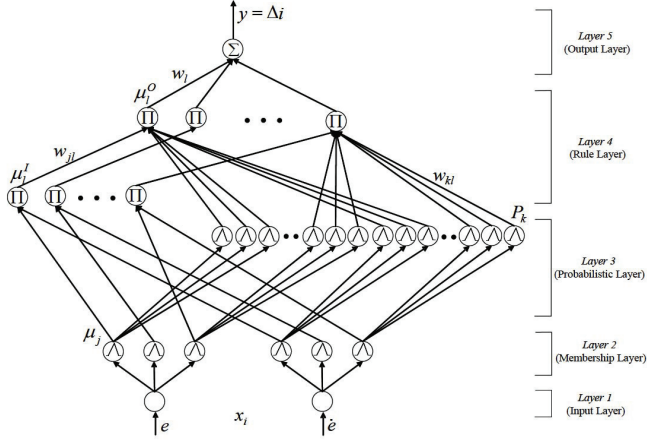


Figure 4. Network structure of PFNN.

A. Probabilistic Fuzzy Neural Network

The architecture of the PFNN including the input layer (layer 1), the membership layer (layer 2), the probabilistic layer (layer 3), the rule layer (layer 4) and the output layer (layer 5) with two inputs and one output is shown in Fig. 4. Moreover, the signal propagation and the basic function of each layer are introduced in the following:

1) *Input layer(layer1)*: In the input layer, the node input and the node output are represented as:

$$x_i(N) = e_i(N), \quad i = 1, 2 \quad (17)$$

where x_i represents the i th input to the input layer; N represents the N th iteration. In this study, the inputs of the PFNN are $e_1(N) = e$ and $e_2(N) = \dot{e}$, which are the tracking error and its derivative, respectively.

2) *Membership layer(layer2)*: In the membership layer, the receptive field function is usually a Gaussian function in FNN. In order to reduce the computational requirements, a triangular function $f_m(x_i)$ is selected as the receptive field function. The equations of the triangular function $f_m(x_i)$ are in the following:

$$\mu_j(x_i) = f_m(x_i) = \begin{cases} 0 & \text{if } x_i \geq m_j + \sigma_j, x_i \leq m_j - \sigma_j \\ \frac{x_i - m_j + \sigma_j}{\sigma_j} & \text{if } m_j - \sigma_j < x_i \leq m_j \\ \frac{-x_i + m_j + \sigma_j}{\sigma_j} & \text{if } m_j < x_i \leq m_j + \sigma_j \end{cases} \quad (18)$$

$i = 1, 2, \quad j = 1, 2, \dots, 6$

where $\mu_j(x_i)$ is the output of the j th node of the i th input variable; m_j is the center of the triangle; σ_j is the center's width of the triangle.

3) *Probabilistic layer(layer3)*: In the probabilistic layer, the receptive field function is usually a Gaussian function. For the same reason in membership layer, another triangular function $f_p(\mu_j)$ is selected as the receptive field function and its equations are as follows:

$$P_k(\mu_j) = f_p(\mu_j) = \begin{cases} 0 & \text{if } \mu_j \geq m_k + \sigma_k, \mu_j \leq m_k - \sigma_k \\ \frac{\mu_j - m_k + \sigma_k}{\sigma_k} & \text{if } m_k - \sigma_k < \mu_j \leq m_k \\ \frac{-\mu_j + m_k + \sigma_k}{\sigma_k} & \text{if } m_k < \mu_j \leq m_k + \sigma_k \end{cases} \quad (19)$$

$k = 1, 2, \dots, 18$

where $P_k(\mu_j)$ is the output of the k th node of the j th input variable; m_k is the center of the triangle; σ_k is the center's width of the triangle.

4) *Rule layer(layer4)*: In the rule layer, each node corresponds to a rule in the knowledge base. In the Mamdani inference, the node itself performs the t-norm operation (product operation) to obtain the inference set according to the rules as shown in (20). The probabilistic information is processed using the Bayes' theorem [1,3] in consideration of the group of fuzzy grade being independent variables as shown in (21). Therefore, the node input and the node output are represented as:

$$\mu_l^I = \prod_j w_{jl} \mu_j \quad (20)$$

$$P_l^I = \prod_k w_{kl} P_k \quad (21)$$

$$\mu_l^O = \mu_l^I P_l^I \quad l = 1, 2, \dots, 9. \quad (22)$$

where μ_l^I and P_l^I are the input of rule layer; w_{jl} is the connective weight between the membership layer and the rule layer which is set to be 1; w_{kl} is the connective weight between the probabilistic layer and the rule layer which is also set to be 1; μ_l^O is the output of the rule layer.

5) *Output layer(layer5)*: In the output layer, the node input and the node output are represented as:

$$\Delta i = y(N) = \sum_{l=1}^9 w_l \mu_l^O \quad (23)$$

where $\Delta i = y(N)$ is the output of the PFNN; w_l is the connective weight between the rule layer and the output layer.

B. On-Line Learning Algorithm Using Delta Adaptation Law

The central part of the learning algorithm for the PFNN

concerns how to recursively obtain a gradient vector in which each element in the learning algorithm is defined as the 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 BP learning rule. To describe the on-line learning algorithm of the PFNN using supervised gradient decent method, first the energy function E is defined as:

$$E = \frac{1}{2}(\omega_r^* - \omega_r)^2 = \frac{1}{2}e^2 \quad (24)$$

Then the learning algorithm is described as follows:

1) *Layer5*: The error term to be propagated is given by:

$$\delta_o = -\frac{\partial E}{\partial y(N)} = -\frac{\partial E}{\partial \omega_r} \frac{\partial \omega_r}{\partial y(N)} \quad (25)$$

and the weights are updated by the amount:

$$\Delta w_l = -\eta_l \frac{\partial E}{\partial w_l} = -\eta_l \frac{\partial E}{\partial y(N)} \frac{\partial y(N)}{\partial w_l} = \eta_l \delta_o \mu_l^o \quad (26)$$

where the factor η_l is the learning rate. The connective weight w_l is updated according to the following equation:

$$w_l(N+1) = w_l(N) + \Delta w_l \quad (27)$$

2) *Layer4*: In this layer, the error terms to be propagated are given by:

$$\delta_l = -\frac{\partial E}{\partial \mu_l^o} = -\frac{\partial E}{\partial y(N)} \frac{\partial y(N)}{\partial \mu_l^o} = \delta_o w_l \quad (28)$$

3) *Layer2*: The error terms to be propagated are given by:

$$\delta_j = -\frac{\partial E}{\partial \mu_j} = -\frac{\partial E}{\partial y(N)} \frac{\partial y(N)}{\partial \mu_l^o} \frac{\partial \mu_l^o}{\partial \mu_j} = \sum_l \delta_l P_l^j \quad (29)$$

By using of the chain rule, the update laws of center and center's width of the triangle are

$$\begin{aligned} \Delta m_j &= -\eta_2 \frac{\partial E}{\partial m_j} = -\eta_2 \frac{\partial E}{\partial y(N)} \frac{\partial y(N)}{\partial \mu_l^o} \frac{\partial \mu_l^o}{\partial \mu_j} \frac{\partial \mu_j}{\partial m_j} \\ &= \begin{cases} -\eta_2 \delta_j \frac{1}{\sigma_j} & \text{if } m_j - \sigma_j < x_i \leq m_j \\ \eta_2 \delta_j \frac{1}{\sigma_j} & \text{if } m_j < x_i \leq m_j + \sigma_j \end{cases} \quad (30) \end{aligned}$$

$$\begin{aligned} \Delta \sigma_j &= -\eta_3 \frac{\partial E}{\partial \sigma_j} = -\eta_3 \frac{\partial E}{\partial y(N)} \frac{\partial y(N)}{\partial \mu_l^o} \frac{\partial \mu_l^o}{\partial \mu_j} \frac{\partial \mu_j}{\partial \sigma_j} \\ &= \begin{cases} \eta_3 \delta_j \frac{m_j - x_i}{(\sigma_j)^2} & \text{if } m_j - \sigma_j < x_i \leq m_j \\ \eta_3 \delta_j \frac{x_i - m_j}{(\sigma_j)^2} & \text{if } m_j < x_i \leq m_j + \sigma_j \end{cases} \quad (31) \end{aligned}$$

where the factors η_2 and η_3 are the learning rates. The center of the triangle m_j and center's width of the triangle σ_j are updated according to the following equation:

$$m_j(N+1) = m_j(N) + \Delta m_j \quad (32)$$

$$\sigma_j(N+1) = \sigma_j(N) + \Delta \sigma_j \quad (33)$$

In order to reduce the computation load, in layer 3, the center of the triangle m_k and the center's width of the triangle σ_k are constants where m_k is set to be 0.4 for $k = 1, 4, 7, 10, 13, 16$, 0.5 for $k = 2, 5, 8, 11, 14, 17$, and 0.6 for $k = 3, 6, 9, 12, 15, 18$; σ_k is set to be 1. Furthermore, the exact calculation of the Jacobian of the system, $\partial \omega_r / \partial y(N)$, is difficult to be determined due to the unknown dynamics of the control system. To overcome this problem, a delta adaptation law is adopted as follows [8]:

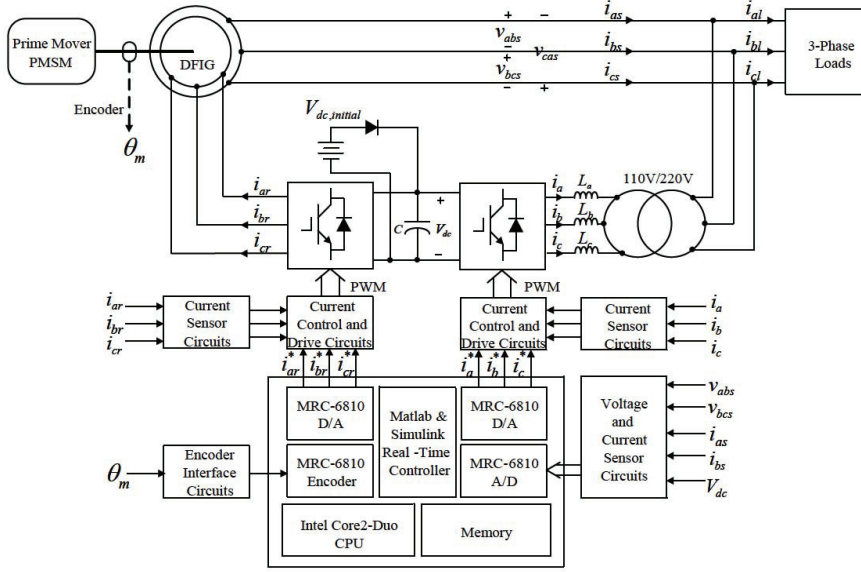
$$\delta_o \equiv e(N) + \dot{e}(N) \quad (34)$$

Moreover, the on-line learning algorithm for the PFNN controller of the stator side converter is the same as PFNN controller of the rotor side converter only $V_n(N)$ is replaced by $V_{dc}(N)$ in related derivations.

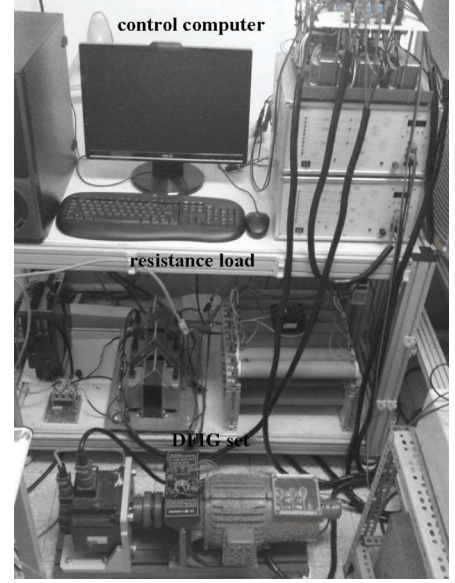
IV. DESIGN AND EXPERIMENTATION

A. Design

The computer controlled DFIG system is shown in Fig. 5. The block diagram of the personal computer (PC)-based control computer for the PFNN controlled DFIG system is shown in Fig. 5(a), where the inductor in the stator side converter is 3mH and the DC-link capacitor is 2820 μ F. The field-oriented control algorithms, synchronization algorithm and the PFNN controllers are realized in a PC. Three-phase current commands are sent to both the rotor side converter and stator side converter using digital/analog (D/A) converters. The current-controlled VSI SPWM converters are implemented using an intelligent power modules (PM30CSJ060) manufactured by Mitsubishi Co. with a switching frequency of 10kHz. Matlab & Simulink Real-Time Control package is adopted for the implementation of the proposed algorithms and

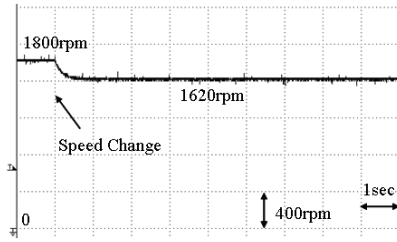


(a)

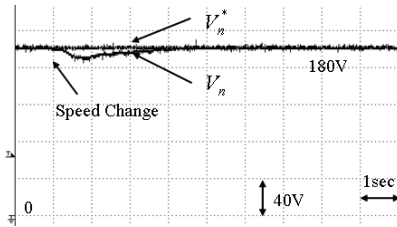


(b)

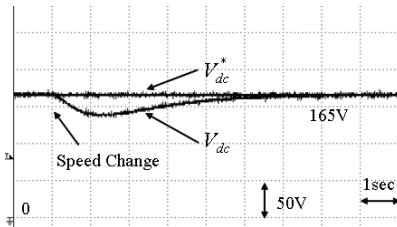
Figure 5. Computer controlled DFIG system. (a) Computer control system. (b) Photo of DFIG set, resistance load and control computer.



(a)

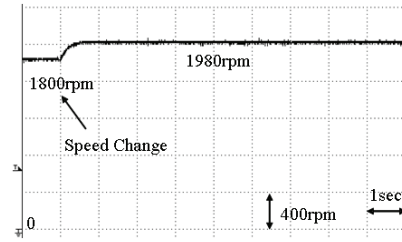


(b)

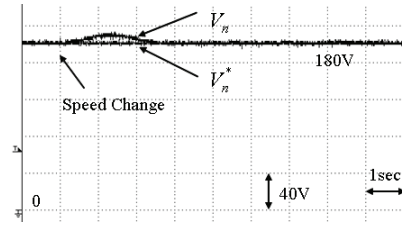


(c)

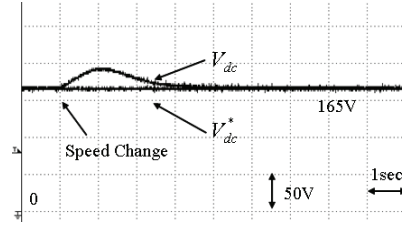
Figure 6. Experimental results of PI controlled DFIG system from 1800rpm to 1620rpm. (a) Speed response. (b) Magnitude response of phase A voltage. (c) Magnitude response of DC-link voltage.



(a)



(b)



(c)

Figure 7. Experimental results of PI controlled DFIG system from 1800rpm to 1980rpm. (a) Speed response. (b) Magnitude response of phase A voltage. (c) Magnitude response of DC-link voltage.

the fix-step solver in the Real-Time Control package is set to be 0.5ms. In the program, first, the v_{abs} , v_{bcs} , V_{dc} , i_{as} , i_{bs} signals are obtained via A/D converters and the θ_m signal is

obtained via encoder interface. Then, both the rotor side and stator side converters are controlled using either PI controller or PFNN controller to generated i_{ar}^* , i_{br}^* , i_{cr}^* and i_a^* , i_b^* , i_c^* ,

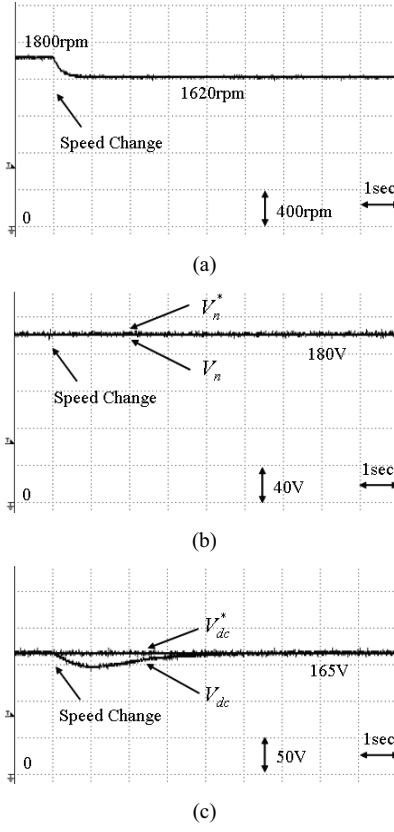


Figure 8. Experimental results of PFNN controlled DFIG system from 1800rpm to 1620rpm. (a) Speed response. (b) Magnitude response of phase *A* voltage. (c) Magnitude response of DC-link voltage.

respectively. The generated current commands are sent to the analog current controllers through D/A converters to generate the pulse-width-modulation (PWM) signals for both the converters. The initial DC-link voltage $V_{dc,initial}$ is also necessary for the stand-alone applications of the DFIG system as shown in Fig. 5(a) to provide the initial rotor excitation. Furthermore, the photo of DFIG set, resistance load and control computer is shown in Fig. 5(b). In addition, in the DFIG set, the prime mover is a field-oriented controlled permanent magnet synchronous motor (PMSM).

B. Experimentation

The adopted DFIG is a three-phase Y-connected four-pole 1Hp 220V/3.5A 1700rpm type. The parameters of the DFIG at the nominal condition are:

$$R_s = 2.85\Omega, R_r = 2.34\Omega, L_s = 0.1967H, L_r = 0.1970H \quad (35)$$

Moreover, the prime mover is a three-phase Y-connected four-pole 3.5kW 120V/26.4A 2000rpm type PMSM. Furthermore, sub-synchronous control 1620rpm ($s=0.1$), synchronous control 1800rpm ($s=0$) and super-synchronous control 1980rpm ($s=-0.1$) three operating conditions all with 484W 3-phase load are test.

To emulate the variation of the wind speed, some experimental results using both the PI and PFNN control are

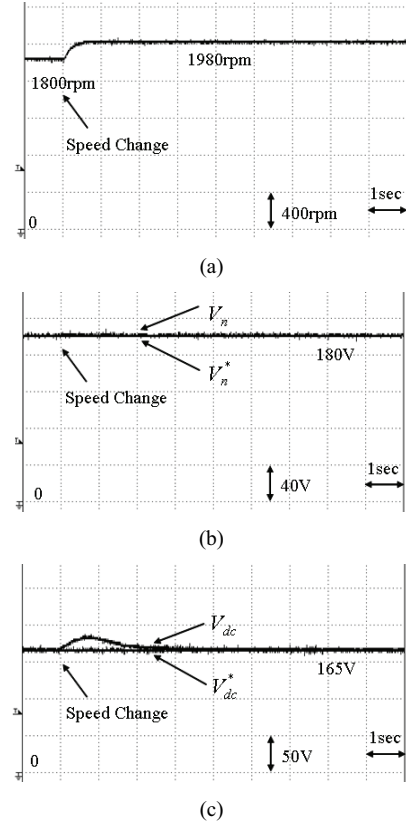


Figure 9. Experimental results of PFNN controlled DFIG system from 1800rpm to 1980rpm. (a) Speed response. (b) Magnitude response of phase *A* voltage. (c) Magnitude response of DC-link voltage.

provided to further test the transient control performance of both the PI and PFNN controlled DFIG system. The variation of the rotor speed, the responses of the magnitude of phase *A* stator voltage V_n and DC-link voltage V_{dc} from 1800rpm to 1620rpm using the PI controller are shown in Figs. 6(a), 6(b) and 6(c). Moreover, the variation of the rotor speed, the responses of the magnitude of phase *A* stator voltage V_n and DC-link voltage V_{dc} from 1800rpm to 1980rpm using the PI controller are shown in Figs. 7(a), 7(b) and 7(c). Furthermore, the variation of the rotor speed, the responses of the magnitude of phase *A* stator voltage V_n and DC-link voltage V_{dc} from 1800rpm to 1620rpm using the PFNN controller are shown in Figs. 8(a), 8(b) and 8(c). In addition, the variation of the rotor speed, the responses of the magnitude of phase *A* stator voltage V_n and DC-link voltage V_{dc} from 1800rpm to 1980rpm using the PFNN controller are shown in Figs. 9(a), 9(b) and 9(c). From the experimental results, compared to the PI controlled DFIG system, excellent magnitudes tracking responses of the phase voltage can be obtained for the PFNN controlled DFIG system. However, the improvement of the responses of the DC-link voltage is limited due to the requirement of the stable phase voltage for the construction of the stable DC-link voltage.

V. CONCLUSION

This study has successfully demonstrated the applications of a PFNN controller on both the rotor side and stator side converters for the voltage magnitudes tracking control of an intelligent controlled DFIG system. First, the dynamic model of a field-oriented control DFIG system via the rotor side converter was introduced. Then, the control methodology of the stator side converter and the synchronization method to the stator were both introduced. Moreover, the network structure, online learning algorithm and convergence analyses of the PFNN were introduced in detail. The PFNN was proposed to control both the magnitudes of phase *A* stator voltage in the rotor side converter and DC-link voltage in the stator side converter at sub-synchronous, synchronous and super-synchronous conditions for stand-alone applications. Furthermore, the effectiveness of the proposed control scheme has been confirmed via experimentation. From the experimental results, the control performance of the proposed PFNN controlled DFIG system is robust with regard to different operating conditions.

REFERENCES

- [1] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065-1076, September 1962.
- [2] A. H. Meghdadi and M.-R. Akbarzadeh-T, "Probabilistic fuzzy logic and probabilistic fuzzy systems," *IEEE International Conf. Fuzzy Syst.*, Melbourne, Australia, 2001, pp. 1127-1130.
- [3] Z. Liu and H. X. Li, "A probabilistic fuzzy logic system for modeling and control," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 6, pp. 848-859, December 2005.
- [4] R. Pena, J. C. Clare, and G. M. Asher, "Doubly fed induction generator using back-to-back PWM converters and its application to variable-speed wind-energy generation," *IEE Proc.-Electr. Power Appl.*, vol. 143, no. 3, pp. 231-241, May 1996.
- [5] D. Forchetti, G. Garcia, and M. I. Valla, "Vector control strategy for a doubly-fed stand-alone induction generator," *IEEE Annual Conf. Industrial Electronics Society*, pp. 991-995, November 2002.
- [6] G. Iwanski and W. Koczara, "DFIG-based power generation system with UPS function for variable-speed applications," *IEEE Trans. Ind. Electron.*, vol. 55, no. 8, pp. 3047-3054, August 2008.
- [7] J. Yao, H. Li, and Z. Chen, "An improved control strategy of limiting the DC-link voltage fluctuation for a doubly fed induction wind generator," *IEEE Trans. Power Electron.*, vol. 23, no. 3, pp. 1205-1213, May 2008.
- [8] F. J. Lin, R. J. Wai, and C. C. Lee, "Fuzzy neural network position controller for ultrasonic motor drive using push-pull DC-DC converter," *IEE Proc.- Control Theory Appl.*, vol. 146, no. 1, pp. 99-107, January 1999.