INTERNATIONAL JOURNAL OF
ENERGY AND ENVIRONMENT

Volume 2, Issue 5, 2011 pp.877-886

Journal homepage: www.IJEE.IEEFoundation.org

MPPT control of wind generation systems based on FNN with PSO algorithm

Chih-Ming Hong[1], Whei-Min Lin[1], Chiung-Hsing Chen[2], Ting-Chia Ou[3]

[1] Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan, R.O.C.
[2] Electronic Communication Engineering, National Kaohsiung Marine University, Kaohsiung 81157, Taiwan, R.O.C.
[3] Institute of Nuclear Energy Research, Atomic Energy Council, Taoyuan 32546, Taiwan, R.O.C.

Abstract
This paper presents the design of an on-line training fuzzy neural network (FNN) using back-propagation learning algorithm with particle swarm optimization (PSO) regulating controller for the induction generator (IG). The PSO is adopted in this study to adapt the learning rates in the back-propagation process of the FNN to improve the learning capability. The proposed output maximization control is achieved without mechanical sensors such as the wind speed or position sensor, and the new control system will deliver maximum electric power with light weight, high efficiency, and high reliability. The estimation of the rotor speed is designed on the basis of the sliding mode control theory.

Keywords: Fuzzy neural network (FNN); Wind turbine (WT); Particle swarm optimization (PSO); Induction generator (IG).

1. Introduction

Wind energy conversion system (WECS) can be found in standalone, hybrid, and grid-connected topologies. The system is highly nonlinear, and a nonlinear control strategy is required to regulate the system for optimal generation. Besides various wind speeds, the effectiveness could be further downgraded with WECS uncertainties. A maximum power point tracking (MPPT) control is required to achieve optimum wind energy utilization and maintain the maximal aerodynamic efficiency. In order to implement maximum wind power extraction, the wind turbine generator must be operated in the variable-speed variable-frequency mode.

Artificial neural network (ANN) are particularly useful to implement nonlinear time-varying input-output mapping. In the past, ANN has been applied for various control, identification, and estimation schemes in power electronics and drives. The ANN-based PWM has advantages of fast parallel computation, learning capability, and fault tolerance, which are not possible by standard PWM implementation methods [1]. Once well trained, the ANN controller could replace the conventional controller with the advantages of increased speed of execution and fault tolerance. ANN can be used to identify and control nonlinear dynamic systems because they can approximate a wide range of nonlinear functions to any desired degree of accuracy.

Wind generation can operate in constant speed or variable speed mode by using power electronic converters. Variable speed generation is attractive because of its characteristics to achieve maximum efficiency at all wind velocities. However, this system requires a rotor speed information for vector control purposes. In this paper, we propose a new rotor speed observer with on-line training fuzzy neural network (FNN) [2-4] controller, and an integral switching surface is designed for the sliding mode speed observer to ensure the stability and robustness under noisy environment. The optimal rotor speed is determined by using the estimation technique [5-6]. Those control strategies can enhance the robustness of the system to capture the maximal wind energy without using the wind speed sensor and improve the dynamic performance [7-8]. It can be seen that a proper design of FNN controller with sliding mode speed observer can yield robust performance under parameter variations and is superior to traditional linear control techniques.

2. Wind turbine characteristics

In order to capture the maximal wind energy, it is necessary to install power electronic devices between the WT and the grid where frequency is constant. The input of a wind turbine is the wind and the output is the mechanical power turning the generator rotor. The output mechanical power available from a wind turbine could be expressed as

$$P_m = \frac{1}{2}\pi\rho C_p(\lambda,\beta)R^2V_\omega^3 \tag{1}$$

where ρ is the density of the air (kg/m^3), R is the radius of the blade, β is the pitch angle of the WT blade. V_ω is the wind speed (m/sec), and C_p is called the power coefficient, and is given as a nonlinear function of the parameter λ with

$$\lambda = \frac{\omega_r R}{V_\omega} \tag{2}$$

where ω_r is the rotational speed. Usually C_p is approximated by $C_p = c_1(\beta)\lambda + c_2(\beta)\lambda^2 + c_3(\beta)\lambda^3$, where $c_1(\beta), c_2(\beta)$, and $c_3(\beta)$ are constructive parameters for a given turbine. t the point $\lambda_{\text{opt}}, C_p = C_{p\,max}$, the maximal power can be captured. It can be seen that $C_{p\,max}$, the maximum value for C_p, is a constant for a given turbine. The dynamic performance of WT could be described as [8]

$$J\frac{d\omega_r}{dt} = T_m - B\omega_r - T_e \tag{3}$$

where *J* is the inertia moment of WT, T_e is the electromagnet moment of the generator, and *B* is the friction coefficient.

3. Design of FNN control system based on PSO algorithm

3.1 System configuration

The model of such a system is well described in many books and papers [9]. The proposed configuration of a field-oriented IG drive system is shown in Figure 1 which consists of an IG, a current-controlled PWM voltage source converter (VSC), a field-orientation mechanism, including the coordinate translator, and a speed control loop. Note that a sliding mode flux observer is proposed in this figure above the speed observer block to provide more accurate rotor speed, and a FNN is proposed to find the current control law i_{qs}^*. By using the reference frame theory and the linearization technique, the field-oriented induction generator system can be reasonably represented , in which

$$T_e = K_t i_{qs}^* \tag{4}$$

$K_t = (3n_p/4)(L_m^2/L_r)i_{ds}^*$, and i_{qs}^* is the torque current command generated from the speed controller, and i_{ds}^* is the flux current command.

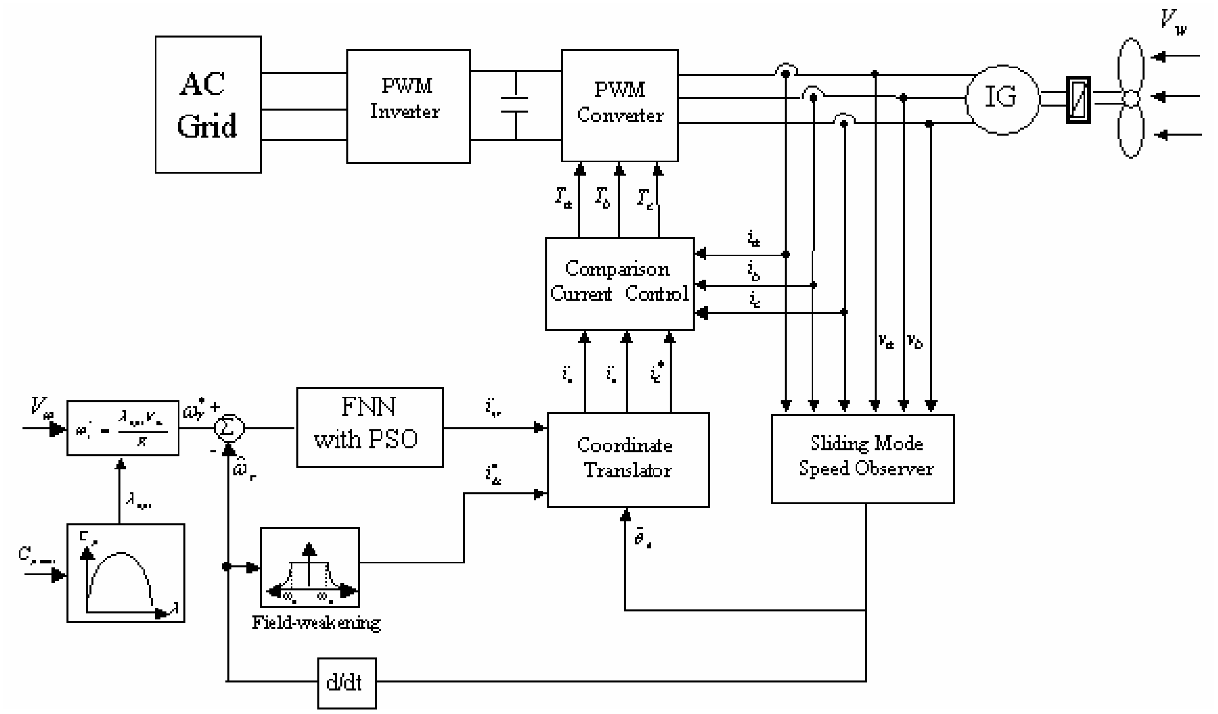


Figure 1. Block diagram of direct field-oriented IG control system

3.2 Fuzzy Neural Network (FNN) with PSO algorithm

A four-layer neural network as shown in Figure 2 is adopted to implement the proposed FNN controller [10-11]. Nodes in layer one are input nodes which represent input linguistic variables. Nodes in layer two are membership modes which act like membership functions. Each membership node is responsible for mapping an input linguistic variable into a probability distribution for that variable. The rule nodes reside in layer three. Altogether, the layer three nodes form a fuzzy rule base. Layer four contains the output variable nodes. This is a simple fuzzy logic system implemented by using a multi-layered feed-forward neural network. The FNN controller is proposed, and the control law is defined as

$$i_{qs}^{*} = u_p \tag{5}$$

where u_p is generated from the FNN controller, and the FNN input is $x_1^{(1)}$ and $x_2^{(1)}$ of the first layer, where $x_1^{(1)} = \omega_r^* - \hat{\omega}_r = e$ and $x_2^{(1)} = \dot{e}$ in this study.

3.2.1 Basic nodes operation

Layer 1: Input layer

The nodes at this layer are used to directly transmit the numerical inputs to the next layer. That is, for the ith node of layer 1, the net input and output are represented as

$$net_i^{(1)} = x_i^{(1)},\ y_i^{(1)} = f_i^{(1)}\left(net_i^{(1)}\right) = net_i^{(1)}\ i = 1{,}2 \tag{6}$$

Layer 2: Membership layer

In this layer, every node performs a membership function. The Gaussian function, a particular example of radial basic functions, is used here as a membership function. We have

$$net_j^{(2)} = -\frac{\left(x_i^{(2)} - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2},$$

$$y_j^{(2)} = f_j^{(2)}\left(net_j^{(2)}\right) = \exp\left(net_j^{(2)}\right) \qquad j = 1,\ldots,n \tag{7}$$

where m_{ij} and σ_{ij} denote respectively, the mean (or center) and the standard deviation, STD, of the Gaussian function in the jth term of the ith input linguistic variable $x_i^{(2)}$ to the node of layer 2. The weights between the input and membership layer are assumed to be unity.

ISSN 2076-2895 (Print), ISSN 2076-2909 (Online)

Layer 3: Rule layer

This layer implements the related links for the term node and rule nodes. In other words, the links in this layer are used to implement the antecedent matching. The matching operation or the fuzzy AND aggregation operation is chosen as the simple PRODUCT operation instead of the MIN operation [3]. Then, for the kth rule node we have

$$net_k^{(3)} = \prod_j w_{jk}^{(3)} x_j^{(3)},$$

$$y_k^{(3)} = f_k^{(3)}\left(net_k^{(3)}\right) = net_k^{(3)} \tag{8}$$

where $x_j^{(3)}$ represents the jth input to the node of layer 3, and $w_{jk}^{(3)}$ is also assumed to be unity.

Layer 4: Output layer

This layer performs the defuzzification to get numerical outputs. The overall net output is a linear combination of the consequence of all rules. If we use the centre-of-area (COA) defuzzification, the net input and output of the jth node in this layer are defined by

$$net_o^{(4)} = \sum_k w_{ko}^{(4)} x_k^{(4)},$$

$$y_o^{(4)} = \frac{f_o^{(4)}\left(net_o^{(4)}\right)}{\sum_k x_k^{(4)}} \tag{9}$$

We need to note that the adopted FNN here must be non-normalized. That is, the operation in layer 4 are defined as

$$y_o^{(4)} = f_o^{(4)}\left(net_o^{(4)}\right) = net_o^{(4)} = u_p \tag{10}$$

where the connection weight $w_{ko}^{(4)}$ is strength of the oth output associated with the kth rule, and is designed to be an adaptively learning parameter.

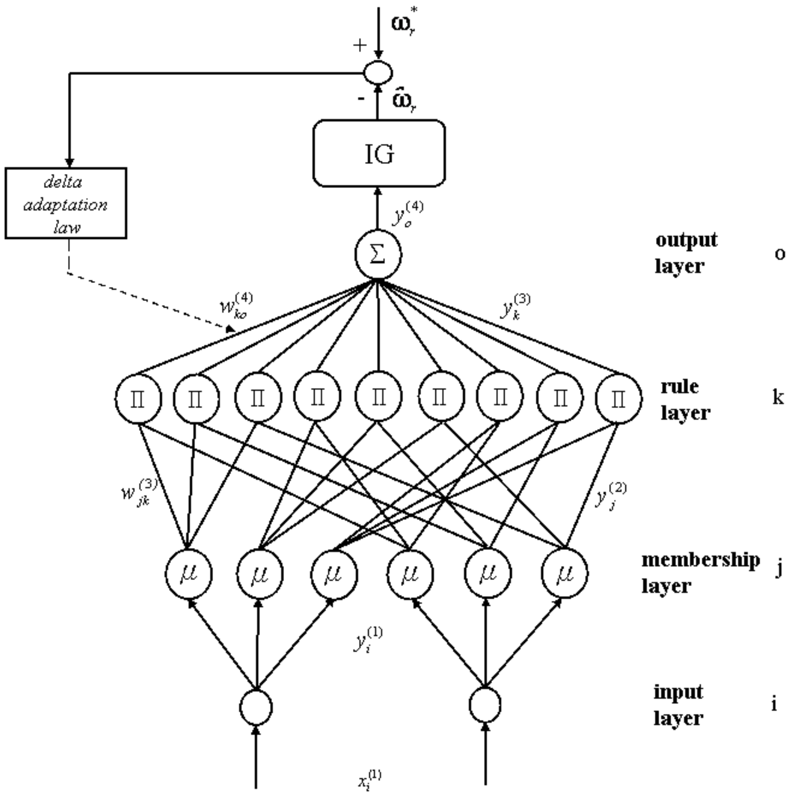


Figure 2. Architecture of the FNN

3.2.2 Supervised learning and training process

Once the FNN has been initialized, a supervised learning law is used to train this system. The basis of this algorithm is gradient descent. The derivation is the same as that of the back-propagation (BP)

algorithm. It is employed to adjust the parameters of the FNN by using the training patterns. By recursive application of the chain rule, the error term for each layer is first calculated. The adaptation of weights to the corresponding layer is then given.
The purpose of supervised learning is to minimize the error function E expressed as

$$E = \frac{1}{2}\left(\omega_r^* - \hat{\omega}_r\right)^2 \tag{11}$$

where ω_r^* and $\hat{\omega}_r$ represent the rotor speed reference and estimated rotor speed of the generator.

<u>Layer 4: Update weight</u>
At this layer, the adjusted weights are $w_{ko}^{(4)}$. The error term to be propagated is given by

$$\delta_o^{(4)} = -\frac{\partial E}{\partial net_o^{(4)}} = \left[-\frac{\partial E}{\partial y_o^{(4)}}\frac{\partial y_o^{(4)}}{\partial net_o^{(4)}}\right] \tag{12}$$

Then the weight $w_{ko}^{(4)}$ is adjusted by the amount

$$\Delta w_{ko}^{(4)} = -\frac{\partial E}{\partial w_{ko}^{(4)}} = \left[-\frac{\partial E}{\partial y_o^{(4)}}\frac{\partial y_o^{(4)}}{\partial net_o^{(4)}}\right]\left(\frac{\partial net_o^{(4)}}{\partial w_{ko}^{(4)}}\right) = \delta_o^{(4)} x_k^{(4)} \tag{13}$$

Hence, the weight is updated by

$$w_{ko}^{(4)}(k+1) = w_{ko}^{(4)}(k) + \eta_w \Delta w_{ko}^{(4)} \tag{14}$$

where $w_{ko}^{(4)}$ is the learning rate for adjusting the parameter η_w

<u>Layer 3:</u>
Since the weights in this layer are unity, none of them is to be modified. Only the error term needs to be calculated and propagated.

$$\delta_k^{(3)} = -\frac{\partial E}{\partial net_k^{(3)}} = \left[-\frac{\partial E}{\partial y_o^{(4)}}\frac{\partial y_o^{(4)}}{\partial net_o^{(4)}}\right]\left(\frac{\partial net_o^{(4)}}{\partial y_k^{(3)}}\frac{\partial y_k^{(3)}}{\partial net_k^{(3)}}\right) = \sum_k \delta_o^{(4)} w_{ko}^{(4)} \tag{15}$$

<u>Layer 2:</u>
The multiplication operation is done in this layer. The error term is computed by

$$\begin{aligned}\delta_j^{(2)} &= -\frac{\partial E}{\partial net_j^{(2)}} = \left[-\frac{\partial E}{\partial y_o^{(4)}}\frac{\partial y_o^{(4)}}{\partial net_o^{(4)}}\frac{\partial net_o^{(4)}}{\partial y_k^{(3)}}\frac{\partial y_j^{(3)}}{\partial net_k^{(3)}}\right]\left(\frac{\partial net_k^{(3)}}{\partial y_j^{(2)}}\frac{\partial y_j^{(2)}}{\partial net_j^{(2)}}\right) \\ &= \sum_k \delta_k^{(3)} y_k^{(3)}\end{aligned} \tag{16}$$

where the subscript k denotes the rule in connection with jth node in layer 2. Then, the adaptive rule for m_{ij} is

$$\Delta m_{ij} = -\frac{\partial E}{\partial m_{ij}} = \left[-\frac{\partial E}{\partial y_j^{(2)}}\frac{\partial y_j^{(2)}}{\partial net_j^{(2)}}\frac{\partial net_j^{(2)}}{\partial m_{ij}}\right] = \delta_j^{(2)} \frac{2\left(y_i^{(1)} - m_{ij}\right)}{\left(\sigma_{ij}\right)^2} \tag{17}$$

and the adaptive rule for σ_{ij} is

$$\Delta \sigma_{ij} = -\frac{\partial E}{\partial \sigma_{ij}} = \left[-\frac{\partial E}{\partial y_j^{(2)}}\frac{\partial y_j^{(2)}}{\partial net_j^{(2)}}\frac{\partial net_j^{(2)}}{\partial \sigma_{ij}}\right] = \delta_j^{(2)} \frac{2\left(y_i^{(1)} - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^3} \tag{18}$$

Thus the updated rules for m_{ij} and σ_{ij}

$$m_{ij}(k+1) = m_{ij}(k) + \eta_m \Delta m_{ij} \tag{19}$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) + \eta_\sigma \Delta \sigma_{ij} \tag{20}$$

where η_m and η_σ are the learning rates for adjusting the parameters m_{ij} and σ_{ij}

3.2.3 Learning rates adjustment using PSO

The PSO is a population-based optimization method first proposed by Kennedy and Eberhart. PSO technique finds the optimal solution using a population of particles. Each particle represents a candidate solution to the problem [12].

Step 1: Initialize random swarm location and velocity

To begin, initial location $R_i^d(N)$ and velocities $v_i^d(N)$ of all particles are generated randomly in whole search space. The generation particles are $R_i^d = [R_i^1, R_i^2, R_i^3]$, where R_i^1, R_i^2, R_i^3 are the FNN learning rates, respectively. The initial pbest of a particle is set by its current position. Then, gbest of a group is selected among the pbests in the group. The random generation of $R_i^d(N)$ initial value ranged as:

$$R_i^d \sim U[\eta_{min}^d, \eta_{max}^d] \tag{21}$$

where η_{min}, η_{max} are the lower and upper bound of the learning rates.

Step 2: Update velocity

During each iteration, every particle in the swarm is updated using (22) and (23). Two pseudorandom sequences $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$ are used to affect the stochastic nature of the algorithm. For all dimensions d, let R_i^d, $Pbest_i^d$, and v_i^d be the current position, current personal best position, and velocity of the jth dimension of the ith particle. The velocity update step is

$$\begin{aligned} v_i^d(N+1) &= wv_i^d(N) + c_1 \cdot r_1 \cdot (Pbest_i^d - R_i^d(N)) \\ &\quad + c_2 \cdot r_2 \cdot (Gbest^d - R_i^d(N)) \end{aligned} \tag{22}$$

Step 3: Update Position

The new velocity is then added to the current position of the particle to obtain its next position

$$R_i^d(N+1) = R_i^d(N) + v_i^d(N+1), i = 1,\ldots,P \tag{23}$$

Step 4: Update pbests

If the current position of a particle is located within the analysis space and does not intrude territory of other gbests, the objective function of the particle is evaluated. If the current fitness is better than the old pbest value, pbest is replaced by the current position. The calculate fitness
value of each particle is select as:

$$FIT = \frac{1}{0.1 + abs(P_m - P_w)} \tag{24}$$

Step 5: Update gbests

In the conventional PSO, gbest is replaced by the best pbest among the particles. However, when such a strategy is applied to multimodal function optimization, some gbests of different groups can be overlapped. To maintain fast convergence rate of PSO, gbest of the group should be selected among the having high fitness value. $Pbest_i^d = [Pbest_1^d, Pbest_2^d, \ldots Pbest_p^d]$

Step 6: Repeat and Check Convergence

Steps 2-5 are repeated until all particles are gathered around the gbest of each group, or a maximum iteration is encountered. The final $Gbest_i^d$ is the optimal learning rate $(\eta_w, \eta_m, \eta_\sigma)$ of FNN.

4. Simulation results

In this section, the sliding mode speed observer as well as the FNN with PSO controller were tested. The proposed FNN with PSO controller is augmented to preserve the desired command tracking response under uncertainties. The optimum rotational speed ω_r^* is obtained for each wind speed V_ω, and used as a reference for the closed loop. Generally the turbine is linked with the generator's shaft using a gearbox, which imposes an additional transform relation in the model. Dynamics of this gearbox are considered unknown in this paper. Examples of the PI controller are used for comparison with the proposed FNN with PSO controller.

4.1 Simulation of variable wind speed

Case 1 : PI controller

Figure 3 shows the performance of a PI controller with sliding mode speed observer. As it can be seen, the actual speed tracking error is high.

Case 2 : FNN with PSO controller

The wind profile is tested with a 5 *msec* sampling time for the wind velocity, with the wind profile a volatile sinusoidal wave. The performance of the FNN with PSO controller and sliding mode speed observer has been investigated emulating wind turbines of different inertia and friction coefficients. Figure 4(a) shows the performance of the FNN with PSO controller and sliding mode speed observer control system. In this case, the sliding mode speed observer tracked the actual speed during the whole wind profile with very small errors. Figure 4(b) shows the tracking error with approximately 0.3rad/sec.

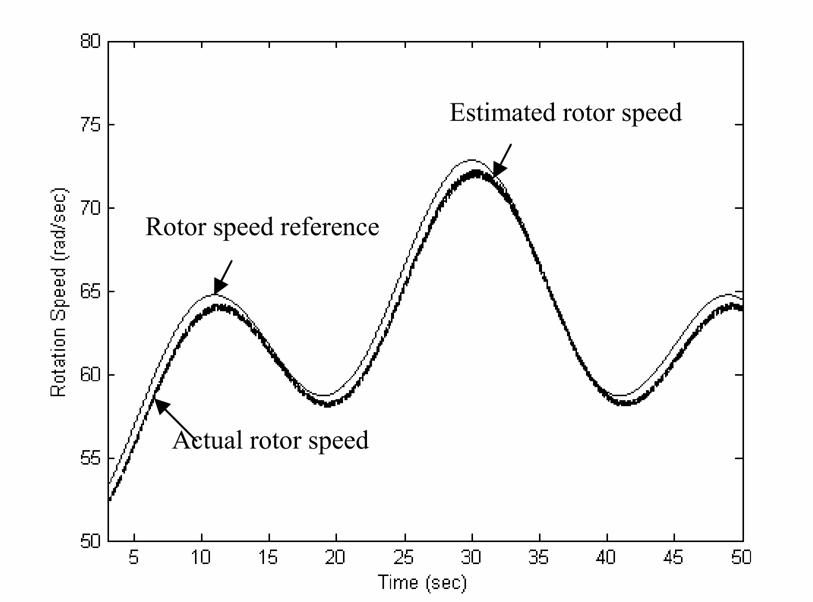
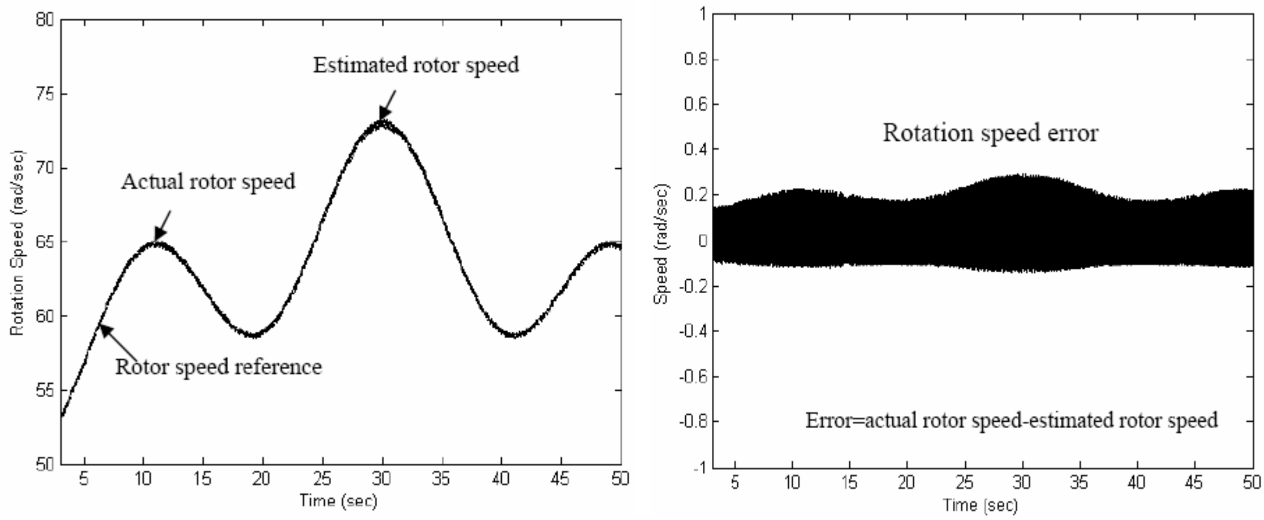


Figure 3. Simulation results of the PI controller speed tracking



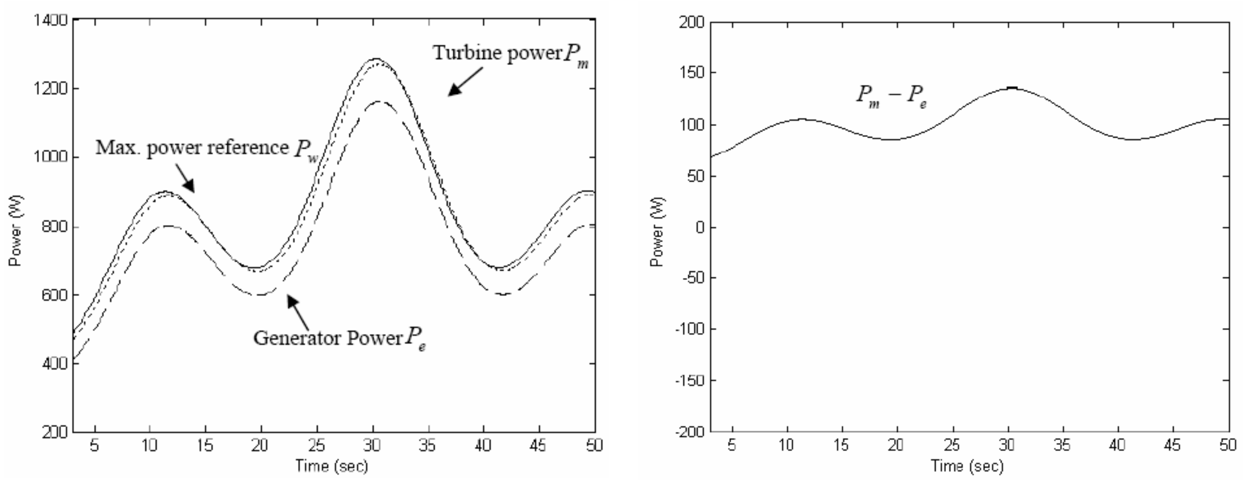
(a) (b)

Figure 4. Simulation results of the FNN with PSO controller speed tracking: (a) The wind tracking, (b) Rotation speed error

4.2 Simulation of the maximum power tracking

<u>Case 1 : PI controller</u>

The verification of maximum power tracking control is shown in Figure 5(a), and Figure 5(b) shows the turbine power P_m and generator power P_e tracking error.

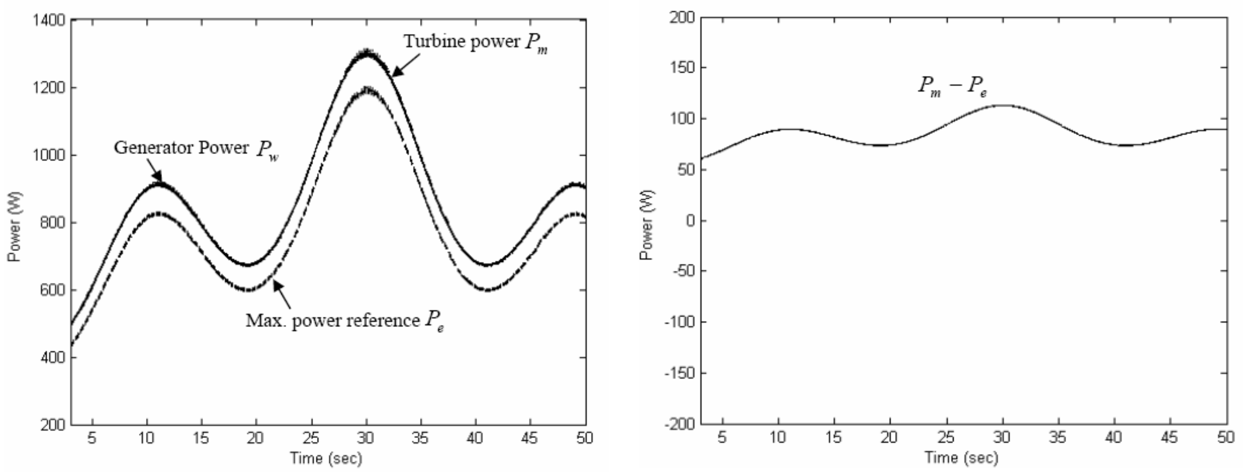


(a) (b)

Figure 5. Simulation results of the PI controller maximum power tracking: (a)The maximum power tracking control signal, (b)Power tracking error

<u>Case 2 : FNN with PSO controller</u>

Maximum power tracking control and the dynamic difference between the turbine power P_m and generator power P_e due to the system inertia and friction are also shown in Figure 6(a) and Figure 6(b). The simulation results show that the wind velocity is well estimated with small errors in both cases. Note that the actual speed is closely tracked by the estimation obtained from the sliding mode speed observer. With the controlled rotor speed, the actual turbine power P_m and the generator power P_e can track the desired P_w closely. The system could capture the maximal wind energy shown in the figures. It shows a robust control performance of the proposed FNN with PSO controller and sliding mode speed observer, both in the wind speed tracking and power regulation.



(a) (b)

Figure 6. Simulation results of the FNN with PSO controller maximum power tracking: (a) The maximum power tracking control signal, (b) Power tracking error

5. Conclusion

This paper presented a new control strategy for IG in a variable speed WECS using sliding mode speed observer to estimate the rotational speed of the IG. We estimate the rotor position from flux linkages using the sliding mode speed observer. The dynamic performance can be used to obtain an accurate estimation of rotational speed not only in steady state but also when fast input changes are applied to the WECS. The algorithms were proposed to cope with the intrinsic nonlinear behavior of wind turbines/generators. The approach, based on a combination of FNN with PSO and a sliding mode speed observer, allowed fast convergence to a simple linear dynamic behavior, even in the presence of parameter changes and model uncertainties, while, the traditional PI controller can not ensure. The proposed FNN with PSO controller and sliding mode observer are successfully implemented in this study for the speed control of WECS.

References

[1] Pinto J. O., Bose B. K., Silva L. E. B, Kazmierkowski M. P. A neural-network-based space-vector PWM controller for voltage-fed inverter induction motor drive. IEEE Trans. Ind. Appl. 2000, 36(6), 1628-1635.

[2] Senjyu T., Sakamoto R., Urasaki N., Funabashi T., Sekine H. Output Power Leveling of Wind Farm Using Pitch Angle Control with Fuzzy Neural Network. IEEE Power Electron. Conf. 2006.

[3] C. T. Lin, C.S. George Lee. Neural Fuzzy Systems. Prentice-Hall, Inc., 1996.

[4] Simoes M. G., Bose B. K., Spiegel R. J. Fuzzy logic-based intelligent control of a variable speed cage machine wind generation system. IEEE Trans. Power Electron. 1997, 12(1), 87-95.

[5] Cardenas R., Pena R., Proboste J., Asher G., Clare J. MRAS Observer for Sensorless Control of Standalone Doubly Fed Induction Generators. IEEE Trans. Energy Convers. 2005, 20(4), 710-718.

[6] Li H., Shi K. L., Mclaren P. G. Neural-Network-Based Sensorless Maximum Wind Energy Capture with Compensated Power Coefficient. IEEE Trans. Ind. Appl. 2005, 41(6), 1548-1556.

[7] Tan K., Islam S. Optimum Control Strategies in Energy Conversion of PMSG Wind Turbine System Without Mechanical Sensors. IEEE Trans. Energy Convers. 2004, 19(2), 392-399.

[8] Rashed M., Goh K. B., Dunnigan M. W., MacConnell P.F.A., Stronach A. F., Williams B. W. Sensorless second-order sliding-mode speed control of a voltage-fed induction-motor drive using nonlinear state feedback. IEE Proc-Electr. Power Appl. 2005, 152(5), 1127-1136.

[9] Bose B.K., Power electronics and AC drive. Prentice-Hall, Englewood Cliffs, New York : John Wiley, 1986.

[10] Lin F. J., Fung R. F., Lin H. H., Hong C. M. A supervisory fuzzy neural network for slider-crank mechanism. Mechatronics. 2001, 11(2),227-250.

[11] Karrari M., Rosehart W., Malik O. P. Comprehensive Control Strategy for a variable Speed Cage Machine Wind Generation Unit. IEEE Trans. Energy Convers. 2005, 20(2), 415-423.

[12] Esmin A. A., Torres G. L., Souza C. Z. A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization. IEEE Trans. Power Systems. 2005, 20(2), 859-866.

Chih-Ming Hong was born in 1972. He received his BS-EE from the Kaohsiung Polytechnic Institute, Kaohsiung , Taiwan, in 1997 and MS-EE degree from the Chung Yuan Christian University, Chung Li, Taiwan, R.O.C., in 1999. He is currently pursuing his Ph.D. degree at National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C.. His research interests include motor servo drives, DSP, control theory applications, and the wind energy conversion systems.
E-mail address: d943010014@student.nsysu.edu.tw

Whei-Min Lin was born in 1954. He received his BS-EE from the National Chao-Tung University, Hsin-Chu, Taiwan, and the MS-EE from the University of Connecticut, Storrs, and his Ph.D. EE from the University of Texas, Arlington, in 1985. He worked at Chung-Hwa Institute for Economic Research, Taiwan, as a visiting researcher after his graduation. He joined Control Data Crop. in 1986, Minneapolis, MN, and worked with Control Data Asia in 1989. He has been with National Sun Yat-Sen university, Kaohsiung, Taiwan, since 1991. Dr. Lin's interests are GIS, Distribution System, SCADA and Automatic Control System. Dr. Lin is a member of IEEE and Tau Beta Pi.

Chiung-Hsing Chen was born in 1961. He received the B.S. degree in electrical engineering from the Feng Chia University, Taichung, Taiwan, in 1983 and M.S. and Ph.D. degree in electrical and computer engineering from the Ohio University, OHIO, USA, in 1994. Currently, he is with National Kaohsiung Marine University, Taiwan, where has been since 1994. His main interests are DSP, SCADA, and intelligent control.

Ting-Chia Ou was born in 1971. He received his BS-EE from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1996 and MS-EE from the National Sun Yat-Sen University, Kaohsiung, Taiwan, in 2002. He is currently pursuing his Ph.D. degree at National Sun Yat-Sen University. He was an Electrical Engineer at Taiwan Railway Administration, Kaohsiung, Taiwan, in 1998. He joined Institute of Nuclear Energy Research in 2008, Tao Yuan, Taiwan, and worked with nuclear power system research. His research interests include Distribution system, Power system operation, Hybrid renewable energy, and Smart grid.