A modified Adaptive Wavelet PID Control Based on Reinforcement Learning for Wind Energy Conversion System Control

M. SEDIGHIZADEH[1,2], A. REZAZADEH[2]

[1]*Faculty of Engineering and Technology, Imam Khomeini International University, Ghazvin, Iran*
[2]*Faculty of Electrical and Computer Engineering, Shahid Beheshti University, G. C.*
Evin 1983963113, Tehran, Iran
m_sedighi@sbu.ac.ir

Abstract—**Nonlinear characteristics of wind turbines and electric generators necessitate complicated and nonlinear control of grid connected Wind Energy Conversion Systems (WECS). This paper proposes a modified self-tuning PID control strategy, using reinforcement learning for WECS control. The controller employs Actor-Critic learning in order to tune PID parameters adaptively. These Actor-Critic learning is a special kind of reinforcement learning that uses a single wavelet neural network to approximate the policy function of the Actor and the value function of the Critic simultaneously. These controllers are used to control a typical WECS in noiseless and noisy condition and results are compared with an adaptive Radial Basis Function (RBF) PID control based on reinforcement learning and conventional PID control. Practical emulated results prove the capability and the robustness of the suggested controller versus the other PID controllers to control of the WECS. The ability of presented controller is tested by experimental setup.**

Index Terms—**Control, Reinforcement, Neural Network, Wavelet, Wind Energy**

I. INTRODUCTION

A nonlinear control strategy is required to set the system in its optimal operation point regarding to nonlinearity and complexity in dynamics of Wind Energy Conversion Systems (WECS). In spite of many improvements in designing Classic and Heuristic nonlinear control for WECS, the PID controller is still widely used for real applications due to its simplicity in implementation and fine-tuning. The main target of designing this controller is to determine three parameters of the PID controller, i.e. proportional gain k_P, integral gain k_i, and derivative gain k_d. One of the most well known methods for tuning of these gains is Ziegler–Nichols method [1]. Other methods are PID self-tuning methods based on the relay feedback technique [2]. However, such tuning methods are suitable for linear systems with time delays and they cannot be used for a WECS since it has complex and nonlinear dynamics. WECS, as a highly nonlinear system, requires a nonlinear control strategy to set the system in its optimal operation point. Thus, in previous works, different methods were introduced based on classic methods and intelligent methods to control of the WECS. Different intelligent approaches have successfully been applied to identify and nonlinearly control the WECS and other plants. For instance, Many authors [3,4,5] surveyed fuzzy logic control, neural network control, expert system control and synthesis intelligent

control methods that is used in the stability, speed control system and maximum-power transfer of WECS. Self-tuning adaptive control approaches are interesting alternatives to control the WECS nonlinear dynamic systems. Developments in the self-tuning adaptive controller design have proved to be useful for a wide class of practical situations [6]. Mayosky and Cancelo [7] used this idea to control the WECS. They proposed a neural-network-based structure consisting of two combined control actions— an RBF (Radial Basis Function) and a supervisory control network-based self-tuning adaptive controller. Sedighizadeh et al. [8 -10] used the idea of Self-tuning control of nonlinear systems using neural network adaptive frame wavelets to identify and control the WECS. They suggested an adaptive PI and PID controller using Rational function with Second-order Poles (RASP1) wavelets for Wind turbine control. Sedighizadeh et al. [11] also suggested an adaptive controller using Morlet wavelets frames neural network for identification and control of WECS. After that, Sedighizadeh et al [12]used the adaptive Radial Basis Function (RBF) PID controller based on reinforcement learning presented by WANG Xue-song et al. [13] to control the WECS. Also Sedighizadeh et al suggested an adaptive PID control based on lyapunov to control the WECS in [14].

In this paper, firstly, the adaptive PID controller based on reinforcement learning presented in [12] is modified, in which the RBF neural network is exchanged by wavelet neural network. Next, the robustness of the proposed adaptive PID controllers is evaluated by adding of noise to parameters of the WECS model. After that, the results are compared with two other controllers. The first one is the controller that is proposed in [12] and the other one is conventional PID controller. Finally, the capability of proposed controller is evaluated using experimental setup.

The mechanism used by reinforcement learning is not similar to supervised learning methods, such as neural network. Indeed, it is an unsupervised on-line learning, which utilizes a trial and error mechanism existing in animals and humans. Initially, a reinforcement-learning agent utilizes the environment in an active manner and afterward, it assesses the utilization results to enhance the controller [13, 15]. One of the most significant reinforcement learning methods is Actor-Critic learning presented by Barto et al., which simultaneously proposes a working method to obtain the optimal action and the expected value [16].

According to this analysis, in this paper, a new wavelet adaptive PID controller, based on reinforcement learning is presented to control the WECS. The Actor-Critic learning method tunes the PID gains adaptively and in an on-line manner. The paper results expose the efficiency and robustness of the new method against other adaptive controllers introduced in this paper to control the WECS.

Details of the Wind Energy Conversion System employed in this simulation are presented in the next section. The adaptive network algorithmic implementation and the controller design steps are described in third Section. Then, in section (4), simulation and practical results are presented and finally, there are conclusions in fifth section.

II. WIND ENERGY CONVERSION SYSTEMS

In this paper, the most common type of wind turbines, i.e. the horizontal-axis type is considered.

The output mechanical power available from a wind turbine is [7]:

$$P = 0.5\rho C_p (V_\omega)^3 A \tag{1}$$

Where, ρ is the air density $[kg/m^3]$, A is the area swept by the blades $[m^2]$, and V_ω is the wind speed $[m/s]$. Power coefficient, C_p is given as a nonlinear function of the parameter $\lambda = \omega R / V_w$, where R is the radius of the turbine $[m]$ and ω is the rotational speed $[rad/\sec]$ of turbine. C_P is approximated as $C_P = a\lambda + b\lambda^2 + c\lambda^3$, where a, b and c are constructive parameters for a specified turbine.

It should be pointed that the maximum value for C_P or $C_{P\max}$, is constant for a specified turbine [7]. This value, when replaced in (1), gives the maximum output power for a specific wind speed. This corresponds to an optimal relationship (λ_{opt}) between ω and V_ω. The torque developed by the wind turbine is:

$$T_l = 0.5\rho \left(\frac{C_p}{\lambda} \right) (V_\omega)^2 \pi R^2 \tag{2}$$

Figure 1 shows the Torque/Speed curves of a typical wind turbine, with V_ω as a parameter. Note that the maximum generated power points ($C_{P\max}$) do not coincide with maximum developed torque points. Optimal performance is achieved when the turbine operates at the $C_{P\max}$ condition. This will be the control objective in this paper.
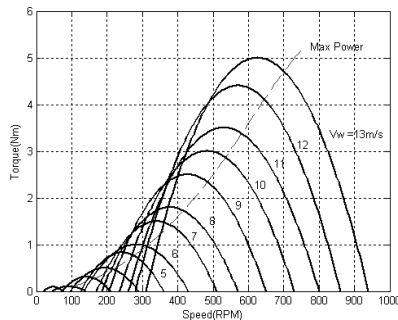

Figure 1. Torque versus Speed curves (solid) of a typical wind turbine. The curve of $C_{P\max}$ is also plotted (dotted).

In this study, the Double Fed Induction Generator (DFIG) is considered. In this generator, slip power is injected to the AC line using a combination of rectifier and inverter, known as a static Kramer drive. Changes on the firing angle (α) of the inverter can control the operation point of the generator in order to develop a resistant torque. This resistant torque places the turbine in its optimum (maximum generation) point. The torque developed by the generator-Kramer drive combination is [7]:

$$T_g = f(\omega, \cos(\alpha)) \tag{3}$$

The dominant dynamics of the whole turbine and generator combination system are those related to the total moment of inertia. Thus, ignoring torsion in the shaft, generator's electric dynamics, and other higher order effects, the approximate dynamic model of the system is:

$$J\omega^\bullet = [T_l(\omega, V_\omega) - T_g(\omega, \alpha)] \tag{4}$$

where J is the total moment of inertia. Regarding (2) and (3), the model of the system is a highly nonlinear model. Moreover, the certain generator parameters are strongly dependent on factors such as temperature and aging. Hence, a nonlinear adaptive control strategy seems very attractive and its objective is to place the turbine in its maximum generation point ($Cp\max$), i.e. ω_{opt}, T_{lopt}, despite wind gusts and changes in the generator parameters. The general form of (4) is $\omega^\bullet = h(\omega, \alpha)$, where h is a nonlinear function accounting for the turbine and generator characteristics. The system is usually designed so that the maximum turbine torque corresponds to 50% to 70% of the peak generator torque. Thus, in this region a simple linearization of the generator expression can be made. Then, the resulting expression after linearization of the generator characteristics for the whole system is [7]:

$$\omega^\bullet = f(\omega) + b\cos(\alpha) \tag{5}$$

Here, f is a nonlinear function of turbine speed and b is a constant.

III. ADAPTIVE PID CONTROLLER BASED ON REINFORCEMENT LEARNING

This controller comprise of two parts: controller architecture and Actor-Critic Learning, which is modified by exchanging of RBF neural network with Wavelet neural network. So these parts are described as follows:

A. Controller architecture

Fig. 2 illustrates the architecture of an adaptive wavelet PID controller based on Actor-Critic learning [12, 13]. The basis of this structure is the idea of incremental PID controller described by Eq. (6).

$$u(t) = u(t-1) + \Delta u(t) = u(t-1) + K(t)x(t) =$$
$$u(t-1) + k_I(t)x_1(t) + k_P x_2(t) + k_D x_3(t) = \tag{6}$$
$$u(t-1) + k_I(t)e(t) + k_P \Delta e(t) + k_D \Delta^2 e(t)$$

Where $x(t) = [x_1(t), x_2(t), x_3(t)] = [e(t), \Delta e(t), \Delta^2 e(t)]^T$ and:

$$e(t) = y_d(t) - y(t)$$
$$\Delta e(t) = e(t) - e(t-1) \tag{7}$$
$$\Delta^2 e(t) = e(t) - 2e(t-1) + e(t-2)$$

where $e(t)$, $\Delta e(t)$ and $\Delta^2 e(t)$ represent the system output error, the first-order difference of error, and the second-order difference of error, respectively; and $K(t) = [k_I(t), k_P(t), k_D(t)]$ is a vector of PID parameters.
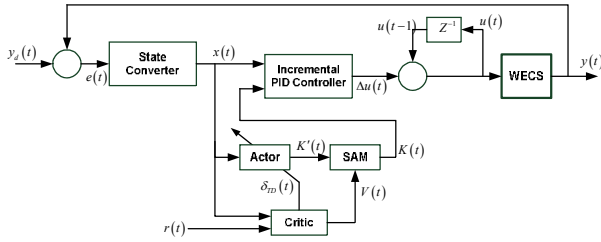

Figure 2. Self-adaptive PID controller based on reinforcement learning.

In Fig.2, the desired and the actual system outputs are presented by $y_d(t)$ and $y(t)$, respectively. The difference between the desired and the actual output, called system error $e(t)$, is imported to the state converter. This converter converts the system error to the system state vector $x(t)$, which is used by the Actor-Critic learning part. Actor-Critic learning structure has three inherent parts. These parts are an Actor, a Critic, and a Stochastic Action Modifier (SAM). The Actor estimates the policy function and maps the current system state vector to the recommended PID parameters $K'(t) = [k'_I(t), k'_P(t), k'_D(t)]$. The proposed PID parameters, $K'(t)$, and the estimated signal from the Critic, $V(t)$, are imported to SAM; and the SAM provides stochastically the actual PID parameters $K(t)$ according to mentioned inputs. The system state vector and an external reinforcement signal from the environment, $r(t)$, (i.e., immediate reward) are received by Critic, which generates a TD error $\delta_{TD}(t)$ (i.e., internal reinforcement signal) and an estimated value function $V(t)$. $\delta_{TD}(t)$ is an essential basis to update the parameters of the Actor and the Critic. The signal $V(t)$ is used by SAM to modify the output of the Actor.

In the course of designing the external reinforcement signal $r(t)$, the impact of the system error and the change rate of the error on control performance should be considered simultaneously. Hence, the external reinforcement signal $r(t)$ is interpreted as [12, 13]:

$$r(t) = \alpha r_e(t) + \beta r_{ec}(t) \tag{8}$$

where α and β are weighted coefficients;

$$
\begin{aligned}
r_e(t) &= \begin{cases} 0 & |e(t)| \le \varepsilon \\ -0.5 & otherwise \end{cases} \\
r_{ec}(t) &= \begin{cases} 0 & |e(t)| \le |e(t-1)| \\ -0.5 & otherwise \end{cases}
\end{aligned}
\tag{9}
$$

and ε is a tolerant error band.

B. Actor-Critic learning based on Wavelet network

The wavelet network employed by the proposed controller is a kind of multi-layer feedforward neural network similar to RBF network. Indeed, the structure of this wavelet network is like the RBF network, but the Morlet wavelet function is used in hidden layer instead of Gaussian function. The concept of wavelet network introduces a

super-wavelet—a wavelet which is a combination of daughter wavelets. Daughter wavelets are simply a dilated and translated version of the original wavelet or mother wavelet. The super-wavelet allows the shape of the wavelet to adapt to a particular problem—a concept that goes beyond adapting the parameters of a fixed shape wavelet. This network has shown good results in controlling nonlinear systems [14]. The wavelet network is a local network in which the output function is well localized in both time and frequency domains. It absorbs advantages such as the multi-resolution of wavelets and the learning of neural network and it can guarantee the good convergence to control nonlinear complex systems [17].

Regarding to the fact that Actor and Critic have the same inputs, the state vector is derived from the environment and just there is a small difference in their outputs. Hence, only one wavelet neural network, as shown in Fig. 3, is used to model the policy function learning of the Actor and the value function learning of the Critic simultaneously. Thus, the Actor and Critic use common inputs and hidden layers of the wavelet network. This structure causes the reduction in the demand for storage space and avoids repeated computation for the outputs of the hidden units; and as a result, it enhances the learning efficiency [12,13].

This network has three layers: Layer1, layer2, and layer3, so called input layer, hidden layer and output layer, respectively. The number of nodes (units) in input layer is equal to the number of inputs. The input vector or system state vector is defined as:

$$x(t) = [x_i | i = 1,2,3] = [e(t), \Delta e(t), \Delta^2 e(t)] \tag{10}$$

where i is an input variable index . Thus, the number of nodes in the input layer is three.
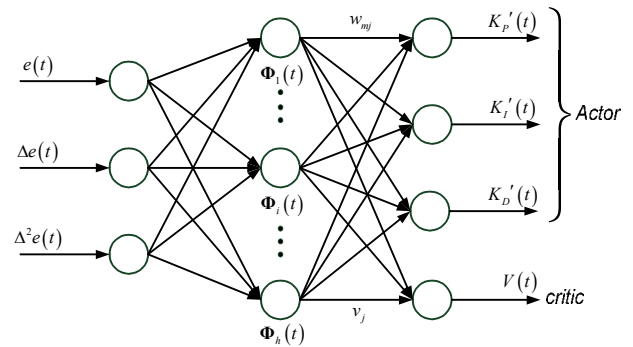

Figure 3. Actor-Critic learning based on wavelet network [13].

The input vector is directly delivered to the hidden layer. In the hidden layer, the Morlet wavelet function is utilized as activation function of the hidden unit of network. The output of the jth hidden unit is:

$$\Phi_j(t) = \prod_{i=1}^{n} \psi_{ij}(\tau_{ij}) \qquad j = 1,...,h \tag{11}$$

where,

$$
\begin{aligned}
\psi_{ij}(\tau_{ij}) &= \cos \omega_0 \tau_{ij} \exp(-0.5\tau_{ij}^2) \\
\tau_{ij} &= \frac{x_i(t) - b_{ij}}{a_{ij}}
\end{aligned}
\tag{12}
$$

$\psi(x)$, which is localized in both time and frequency

domains, is called a Morlet wavelet; and the parameters a_{ij} and b_{ij} are named dilation and translation parameters, respectively. ω_o is the wavelet frequency, h is the number of hidden units, and n is number of inputs. The output layer has two parts: an Actor part and a Critic part. The mth output of the Actor part, $K'_m(t)$, and the value function of the Critic part, $V(t)$, are computed as:

$$K'_m(t) = \sum_{j=1}^{h} w_{mj}(t)\Phi_j(t), \quad m=1,2,3 \tag{13}$$

$$V(t) = \sum_{j=1}^{h} v_j(t)\Phi_j(t) \tag{14}$$

where w_{mj} represents the weight between the jth hidden unit and the mth Actor unit, and v_j indicates the weight between the jth hidden unit and the single Critic unit. In this structure, the number of output nodes is four.

According to [12, 13], the output of the Actor part is not used by PID controller directly. Therefore, a Gaussian noise term η_k is added to the suggested PID parameters, $K'(t)$, which are exported by the Actor. The actual PID parameters, $K(t)$, are denoted as Eq. (13). In this equation, the magnitude of the Gaussian noise is not constant and it depends on the estimated value function $V(t)$. This dependency is described as Eq. (15) and (16).

$$K(t) = K'(T) + \eta_k(0, \sigma_V(t)) \tag{15}$$

where

$$\sigma_V(t) = \frac{1}{1 + \exp(2V(t))} \tag{16}$$

Ii is clear, the important property of Actor-Critic learning is that the policy function is learned by the Actor, and the value function is learned by the Critic using the TD method simultaneously [16]. The temporal difference of the value function among consecutive states in the state transition forms the TD error $\delta_{TD}(t)$ as follows:

$$\delta_{TD}(t) = r(t) + \gamma V(t+1) - V(t) \tag{17}$$

Where $r(t)$ is the external reinforcement reward signal; and γ, which its value is between zero and one, indicates the dependency rate of TD error to the future rewards. The TD error dictates decency of the actual action. Hence, the performance index function of system learning should be expressed as follows.

$$E(t) = \frac{1}{2}\delta_{TD}^2(t) \tag{18}$$

With using the TD error performance index, the gradient descent method, and a chain rule, the weights of Actor and Critic are updated according to the following equations:

$$w_{mj}(t+1) = w_{mj}(t) + \alpha_A \delta_{TD}(t)\frac{K_m(t) - K'_m(t)}{\sigma_V(t)}\Phi_j(t) \tag{19}$$

$$v_j(t+1) = v_j(t) + \alpha_C \delta_{TD}(t)\Phi_j(t) \tag{20}$$

where α_A and α_C are learning rates of Actor and Critic,

respectively.

Since the Actor and the Critic employ the same inputs and the same hidden layers of wavelet network, the translations and the dilations of hidden units are updated once according to the following equations:

$$b_{ij}(t+1) = b_{ij}(t) + \eta_b \delta_{TD}(t)v_j(t)\frac{\partial \psi_{ij}}{\partial b_{ij}}[\prod_{k=1,k\neq i}^{n}\psi_{kj}(\tau_{ki})] \tag{21}$$

where,

$$\frac{\partial \psi_{ij}}{\partial b_{ij}} = \frac{1}{a_{ij}(t)}\Big[\omega_o \sin\omega_o\tau_{ij}\exp(-0.5\tau_{ij}^2) + \tau_{ij}\psi_{ij}(\tau_{ij})\Big] \tag{22}$$

$$a_{ij}(t+1) = a_{ij}(t) + \eta_a \delta_{TD}(t)v_j(t)\frac{\partial \psi_{ij}}{\partial b_{ij}}\tau_{ij}[\prod_{k=1,k\neq i}^{n}\psi_{kj}(\tau_{ki})] \tag{23}$$

In equation (21) to (23), η_b and η_a are learning rates of translation and dilation respectively.

Regarding to the notes described in this section, the overall block diagram of proposed controller can be derived as Fig. 4, which illustrates the whole design steps of presented adaptive PID controller.

IV. SIMULATION RESULTS

Fig. 4 depicts the block diagram of the adaptive PID Controller based on Reinforcement Learning for WECS control while the dynamic of the WECS is described by Eq. (5). The data of simulated WECS is listed as table 1 in the appendix. For this case study, the desired signal $y_d(t)$ is optimal rotor speed ω_{opt}; the actual output $y(t)$ is rotor speed ω; and the control signal $u(t)$ is firing angle of inverter (α). The optimum shaft rotational speed, ω_{opt}, is obtained for each wind speed V_ω and is used as a reference for the closed loop. Note that the wind speed also acts as a perturbation on the turbine's model. The proposed adaptive PID controller and the other controllers are used to track the optimal rotor speed signal. Sampling period is $T_s = 0.0015s$ during the simulation process. PID parameters of the conventional PID controller are set off-line as $k_P = 0.15$, $k_I = 0.55$ and $k_D = 0.005$ using the Ziegler-Nichols tuning rule. The corresponding parameters for the proposed adaptive PID controller are set as: $\alpha = 0.67$, $\beta = 0.47$, $\varepsilon = 0.014$, $\gamma = 0.92$, $\alpha_A = 0.017$, $\alpha_C = 0.014$, $\eta_a = 0.032$, $\eta_b = 0.018$ and $h = 6$. To create of a base line for comparison, the parameters of adaptive PID controller based on RBF network presented in [12] is adjusted similar to proposed controller. Detailed simulation results are illustrated in the following figures. Comparing the results in Figures 5, it can be noticed that the WECS output (i.e., the rotor speed) tracks the desired output more precisely when using the proposed controller and RBF based controller than using the conventional PID controller. Fig 6 shows the firing angle of inverter (input) responses for the presented and conventional controllers. Regarding to figure 6, it is clear that when the new controller is employed, computational efforts are less than the case when the

conventional controller is utilized; and this new controller is more convenient for hardware and software implementations. The system error is shown in Fig. 7. This figure reconfirms the capability of the new controller versus the conventional PID controller in controlling the WECS.
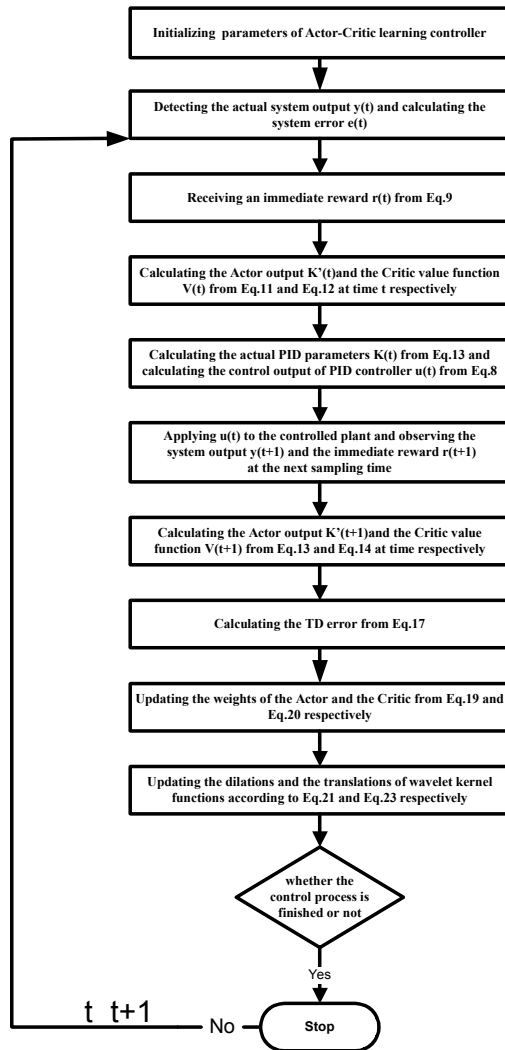
Figure 4. Overall Controller block diagram.

In order to analyze the robustness of the new controller, we suppose that the rotor resistance, R_r, is contaminated by a Gaussian white noise with the mean value equal to zero and the variance equal to 0.01. Figures 8 to 10 show the results related to the noisy condition. In this condition, while the proposed controller follows the reference output fairly, but the conventional PID cannot capture it. In following figures, it is seen the proposed controller and RBF based controller have similar results to track of desired signal.

The simulation results indicate that the proposed adaptive PID controller exhibits perfect control performance and adapts to the parameters changes of the WECS. Therefore, it has the characteristics of being strongly robust and adaptable.
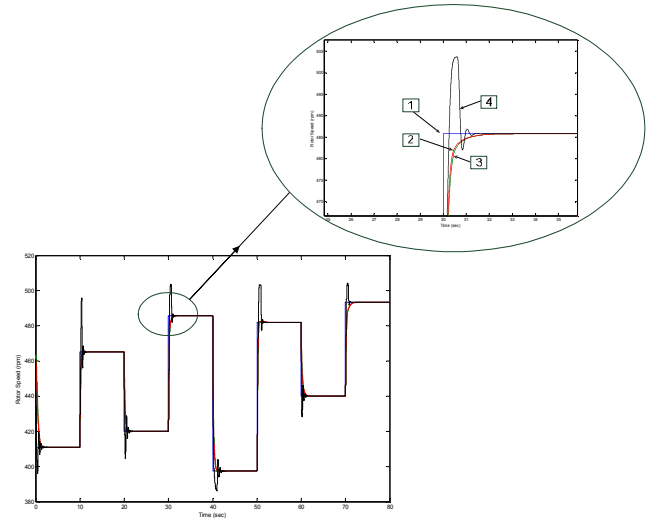
Figure 5. Rotor speed response.

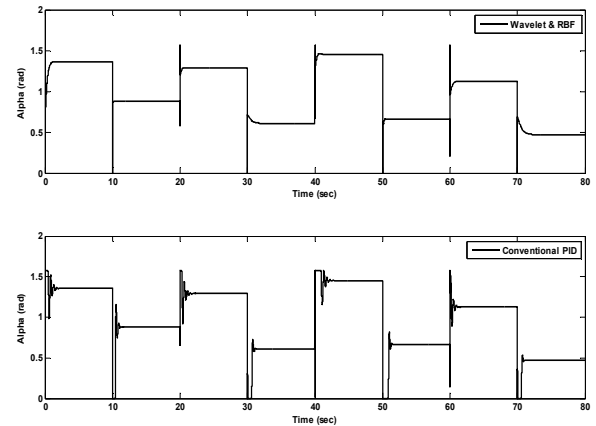(1- Reference signal 2- proposed controller 3- RBF based controller 4- conventional PID)

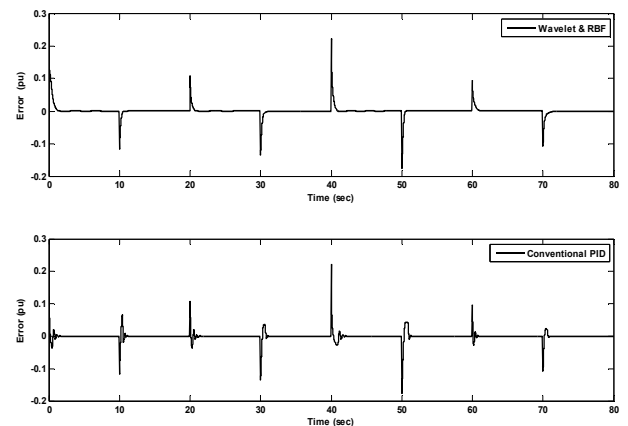**Figure 6.** Firing angle of inverter response. Up: proposed and RBF based PID, Down: conventional PID.

Figure 7. System error, Up: proposed and RBF based PID controller, Down: conventional PID.
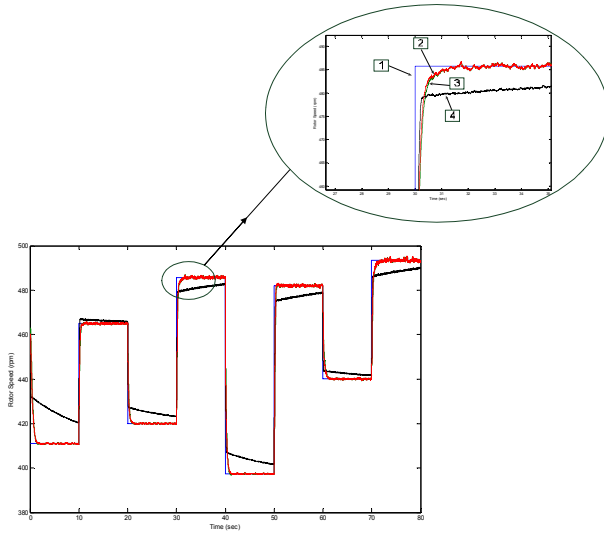
Figure 8. Rotor speed response (with noise in parameter). (1- Reference signal 2- proposed controller 3- RBF based controller 4- conventional PID)
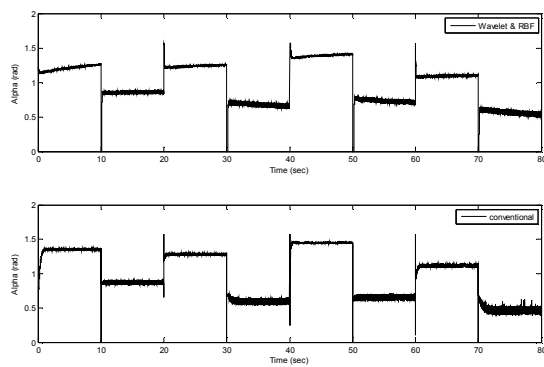

Figure 9. Firing angle of inverter response. Up: proposed and RBF based PID, down: conventional PID (with noise in parameter).
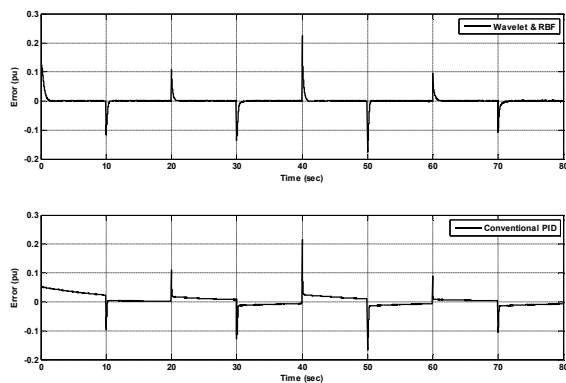

Figure 10. System error, Up: proposed and RBF based PID controller, Down: conventional PID (with noise in parameter).

V. EXPERIMENTAL SETUP

As maintained earlier the performance of the simulated results of proposed control, is proved by the aim of an experimental setup. The schematic of the system is depicted in figure 11.

A DC brushed motor is considered to emulate the wind turbine that acts as the prime mover for the double fed induction generator. The DC driver is used as power amplifier for DC motor. To implement the emulation, the detailed model of WECS that has been defined in this paper

is implemented in the host PC using C++. Then for each wind speed with considering of previous firing angle of static Kramer drive, the rotational speed of DC motor is determined in each sampling time. The driving DC motor forces the double fed induction generator rotational speed to this value. With this emulation technique, the generator rotates at the same speed as that of a generator driven by a real wind turbine. After that, the optimum rotational speed to achieve maximum output power is calculated using host PC for each wind speed and it is sent to FPGA-based external hardware as a reference for controller. The proposed controller is emulated in this FPGA-based hardware, which generates desired firing angle static Kramer Drive until rotational speed of WECS is moved toward optimal speed. The experimental prototype is rated at about 3.5 kW. The experimental rig is shown in fig 11.
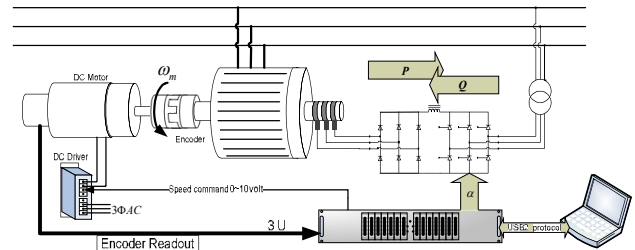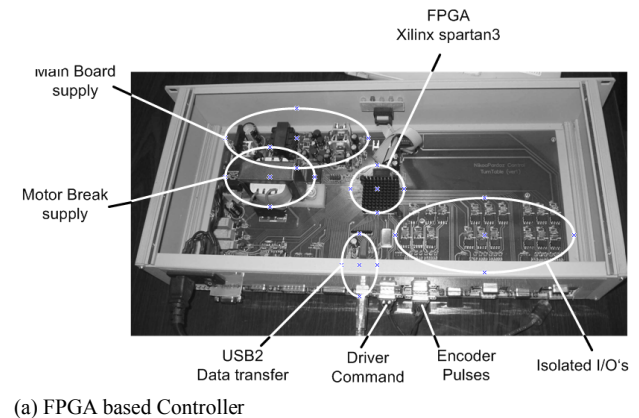

Figure 11. The experimental Setup.

The control box is FPGA (Xilinx Spartan XC2S100PQ208) based and designed to negotiate with windows XP based software by USB2 port via FT245. The FPGA based controller and overall setup is shown in fig 12. Fig 13 shows the performance of the proposed controller when the wind profile of fig. 5 is used in the wind turbine emulator.

(a) FPGA based Controller

(b) Overall setup

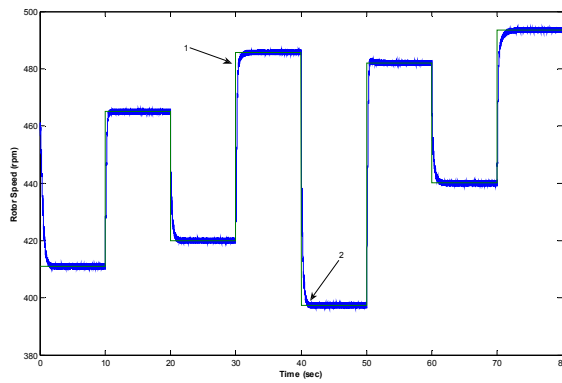**Figure 12.** experimental emulator setup.

Figure 13. Experimental results using a wind turbine emulator 1- reference signal 2- response of proposed controller.

VI. CONCLUSIONS

In his paper, the simulation results denote that the proposed adaptive PID controller can be a suitable candidate for controlling the WECS. It is robust for system disturbances and it shows better results in comparison with a conventional PID controller.

The wavelet function utilization in the hidden layer of the neural network, can guarantee the good convergence of the neural net system. Also sharing the input and the hidden layers of wavelet network by the Actor and Critic can reduce the demand for storage space and can avoid repeated calculations. Regarding to the described subjects, the proposed PID controller can be a convenient alternative to be used in industrial applications.

ACKNOWLEDGMENTS

APPENDIX

TABLE. 1 THE DATA OF SIMULATED WECS [7]

Turbine		
Parameters	**Value**	**Units**
ρ	1.204	Ns^2/m^2
R	0.95	m
J	0.312	Nm^2/rad
a	-0.10453	
b	0.07693	
c	-0.01046	
Generator		
Parameters	**Value**	**Units**
R_f	0.1	Ω
R_s	0.2	Ω
R_r	0.2	Ω
L_{ls}	0.9e-3	h
L_{lr}	0.4e-3	h
n_1	1	
n_2	1	

REFERENCES

[1] K.J. Astrom, B. Wittenmark, Adaptive Control, Addison-Wesley, New York, 1995.
[2] A. Leva, PID autotuning algorithm based on relay feedback, IEE Proceedings-D 140 (5) (1993) 328–37.
[3] Chedid R, Mrad F, Basman M (1999) Intelligent control of a class of wind energy conversion system. IEEE T-EC 14(4):1597–1604
[4] Kanellos FD, Hatziargyriou ND (2002) A new control scheme for variable speed wind turbine using neural networks. IEEE Power Eng Soc Winter Meeting 1(1):360–365
[5] Kyoungsoo Ro,Han-ho Choi (2005) Application of neural network controller for maximum power extraction of a grid-connected wind turbine system. Electr Eng (Archiv Elektrotech) 88(1):45–53
[6] Narendra. K. S., K. Parthasarathy,"Identification and control of dynmical systems using neural networks," IEEE Trans. On neural networks, Vol: 1, No: 1, march 1990, pp: 4-27
[7] Mayosky, M. A., Cancelo, G. I. E., "Direct adaptive control of wind energy conversion systems using gaussian networks," IEEE Transactions on neural networks, 1999, Vol. 10, No.4, pp: 898-906.
[8] Kalantar M, Sedighizadeh M (2004) Adaptive self tuning control of wind energy conversion systems using Morlet mother wavelet basis functions networks. In: 12thMediterranean IEEE conference on control and automation MED'04, Kusadasi
[9] Sedighizadeh M, Kalantar M (2004) Adaptive PID control of wind energy conversion systems using RASP1 mother wavelet basis function networks. IEEE TENCON 2004, Chiang Mai
[10] Sedighizadeh M, et al. (2005) Nonlinear model identification and control of wind turbine using wavenets. In: Proceedings of the 2005 IEEE conference on control applications, Toronto, pp 1057–1062
[11] Sedighizadeh M, Rezazadeh A (2008) Self Tuning Control of Wind Turbine Using Neural Network Identifier. Electr Eng (Archiv Elektrotech). Volume 90, Number 7, Sept. 2008, PP. 479-491.
[12] M. Sedighizadeh, A.Rezazadeh, (2008) Adaptive PID Controller based on Reinforcement Learning for Wind Turbine Control. Proceedings of World Academy of Science, Engineering and Technology (CESSE2008), Cairo, Egypt, Vol. 27 , February 2008, ISSN 1307-6884, PP.257-262
[13] Wang XS, Cheng YH, Sun W (2007). A Proposal of Adaptive PID Controller Based on Reinforcement Learning. J China Univ Mining & Technol .17(1): 0040–0044
[14] Sedighizadeh, M.; Rezazadeh, A.; Khatibi, M.;" A self-tuning PID control for a wind energy conversion system based on the Lyapunov approach**",** 43rd International Universities Power Engineering Conference, 2008. UPEC 2008, 1-4 Sept. 2008, PP:1 - 4
[15] Wang X S, Cheng Y H, Sun W. Q learning based on self-organizing fuzzy radial basis function network. Lecture Notes in Computer Science, 2006, 3971: 607–615.
[16] Barto A G, Sutton R S, Anderson C W (1983). Neuron like adaptive elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man and Cybernetics.13(5): 834–846.
[17] Szu HH, Telfer BA, Kadambe S (1992) Neural network adaptive wavelets for signal representation and classification. Opt Eng 31(9):1907–1916