Output Electrical Power Control of Horizontal Axis Wind Turbine Using Indirect Model Reference Adaptive Neuro Controller

Vahid Azimi[1], Mohammad Bagher Menhaj[2]
1- Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH, USA
Email: vahid.azimii@gmail.com, v.azimi@csuohio.edu (Corresponding Author)
2- Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran
Email: Menhaj@aut.ac.ir

Received: May 2014 Revised: Dec. 2014 Accepted: March 2015

ABSTRACT:
In this paper, we investigated Indirect Model Reference Adaptive Neuro Control (IMRANC), for output electrical power tracking of a nonlinear non-affine Horizontal Axis Wind Turbine (HAWT). Firstly, the nonlinear system is identified by the Nonlinear Autoregressive network with Exogenous inputs (NARX) model that is a recurrent dynamic network. Afterward an IMRANC is designed based on NARX identified model to reach the close loop system in order to get the desired reference model. The MLP networks are applied for both of model and controller subsystems and are then trained by the Marquardt-Levenberg Back-Propagation (LMBP) algorithm while the Tapped Delay Lines (TDL) components are considered over input and feedback paths. Finally, simulation results are presented to validate the effectiveness of the proposed method like robustness and good load disturbance attenuation and accurate tracking, even in the presence of parameter variations due to changing of hydraulic pressure in hydraulic pitch system and also disturbances on the system.

KEYWORDS: IMRANC, NARX model, MLP network, LMBP algorithm, HAWT

1. INTRODUCTION

Among the renewable energy sources available today, wind power is the world's fastest growing. With an annual growth rate in installed wind energy capacity of 30% on average throughout the past 10 years, wind turbines are definitely up and coming. For several reasons wind energy is growing fast: it is cheap, inexhaustible, widely distributed, clean, and climate friendly. Since the dynamics of the HAWT are highly nonlinear and non-affine and may contain uncertain parameters such as the damping ratio and undamped natural frequency of hydraulic pitch system due to changing hydraulic pressure, many efforts have been made in developing control designs to achieve the accurate tracking control of the HAWT. Over the last decade, there have been numerous progresses in the development of controllers for wind turbines as listed below:

1. A. S. Yilmaz *et al*. [1] proposed pitch angle control in wind turbines above the rated wind speed by multi-layer perceptron and radial basis function neural networks.
2. C. Sloth *et al*. [2] designed robust and fault-tolerant linear parameter-varying control of wind turbines.
3. H. Camblong [3] investigated the digital robust control of a variable speed pitch regulated wind turbine for above rated wind speeds.
4. P. F. Odgaard *et al*. [4] used fault tolerant control of wind Turbines -a benchmark model.
5. O. Ognyanova *et al*. [5] presented robust control of a wind turbine.
6. K. E. Johnson *et al*. [6] proposed control of variable speed wind turbines.

Furthermore other different dissertations of wind turbine control have been published in [7-10].

The main contribution of this research is indirect model reference adaptive neuro control of Horizontal Axis Wind Turbine (HAWT). In this paper the problem of output electrical power control of a wind turbine, which possesses not only parameter uncertainties but also external disturbances, is considered. The HAWT is a rapidly growing source of alternative energy that presents numerous control problems. One of these problems is the HAWT dynamics that are extremely nonlinear. Some papers have recently used a T-S fuzzy model which can be utilized to approximate global behavior of highly complex nonlinear systems [11-19]. Whereas, in this research a recurrent NARX model will be designed for dynamic model of the HAWT. This study concentrates on developing a controller that can track a desired electrical power based on a recurrent NARX model of the HAWT and using the MLP network despite nonlinear system uncertainties. The NARX model is a mighty class of models which has been approved that they are well suited for modeling nonlinear systems [20-22].

First we consider a NARX model on behalf of the nonlinear real plant. Now if the error between outputs of original system and estimated model is small, then applying a control signal to the approximate model would cause a similar behavior of the real system. The IMRANC scheme then uses a NARX neural network to identify unknown parameters and a global neural network feedforward controller is designed and trained using LMBB algorithm, see [23-25]. Finally, simulation results show that the proposed method can effectively meet the performance requirements like robustness and good load disturbance rejection, good tracking and fast transient responses of the HAWT.

The paper is organized as follows. The wind turbine model is presented in Section II and the problem statement is described in Section III. In section IV, the proposed control scheme is designed. Simulation results of the closed-loop system with the proposed controller are presented in Section V and finally the paper is concluded in Section VI.

2. WIND TURBINE MODEL

The majority of classic HAWT are three-axis horizontal-axis wind turbines and as depicted in Fig. 1, structure of them consists of tower, nacelle, hub and blade.

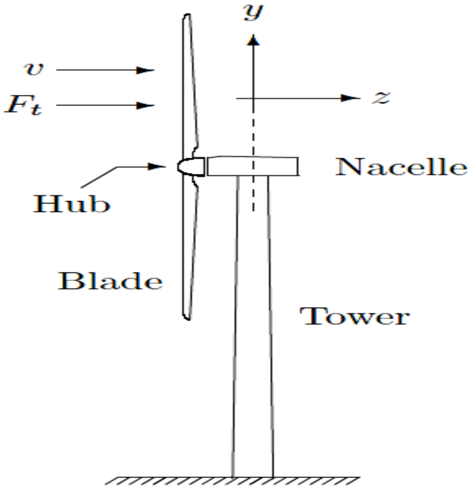


Fig.1. A horizontal-axis wind turbine (HAWT)

The tower provides elevation for the rotor and with the obtained increased height, there would be more stable wind flows and height allows rotor dimensions to increase. The nacelle mounted on the tower contains the key components of the wind turbine, including the gearbox, the electrical generator, and the main shaft providing the mechanical interface to the hub carrying the blades. Hub is at the end of the nacelle. The blades are attached to the hub using bearings in order to allow the blades to be rotated around their own axis. The blade angle is referred to as the pitch angle. The blades of the rotor are influenced by the passing wind and kinetic energy is transferred via the hub to the rotor shaft. When the wind interacts with the blade aerodynamics, it will emerge a torque on the rotor, which is transferred to the generator via the drive train. The drive train consists of a low-speed shaft connecting the hub to the gearbox, and a high-speed shaft connecting the gearbox to the generator. The HAWT dynamics can be described as joined block models according to Fig. 2 that show the interaction of the signals of the wind turbine system [7- 10]. Each of the different blocks in the block diagram of the full model will be explained in following subsections.

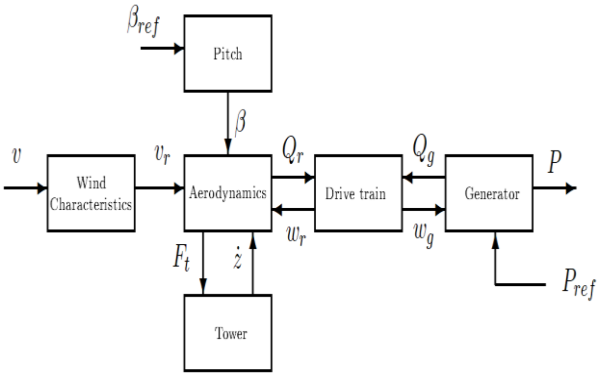


Fig. 2. Block diagram of the full model

Wind Model

Wind is chaotic in nature. Thus, we describe the wind speed v as a mean wind speed $\bar v$ that perturbed by a turbulent wind speed $\tilde v$

$$v = \bar v + \tilde v \tag{1}$$

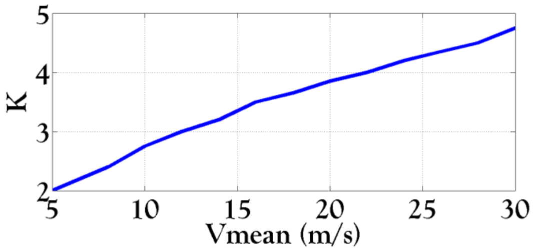
Note that the wind speed v_r seen by the rotor plane is the incoming wind speed v superimposed by the nacelle velocity $\dot z$ resulting from the tower being deflected by the wind

$$v_r = v - \dot z \tag{2}$$

The state-space description of turbulence process can be expressed as follows

$$\ddot{\tilde v} = -\frac{1}{\tau_1\tau_2}\tilde v - \frac{\tau_1+\tau_2}{\tau_1\tau_2}\dot{\tilde v} + \frac{k}{\tau_1\tau_2}\delta \tag{3}$$

Where δ is a zero mean white noise with unit variance, and k, τ_1 and τ_2 depend on the mean wind speed that is shown in Fig.3. According to Fig. 3 an increase in the mean wind speed make happens an increase in the k, but increase in the mean wind speed contributes to an decrease in τ_1 and τ_2.



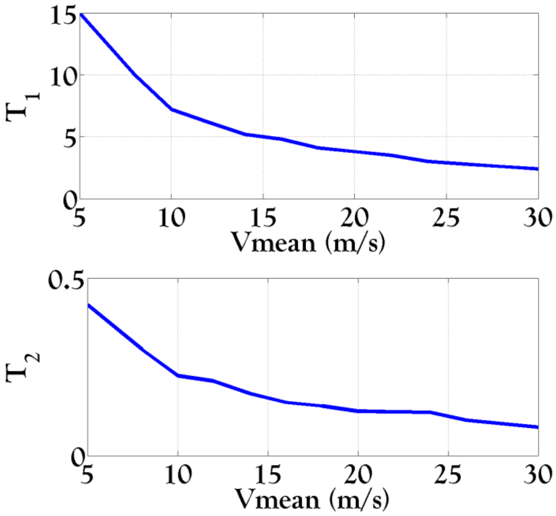


Fig.3. Parameters in the turbulence model k, τ_1 and τ_2 as the functions of the mean wind speed

Aerodynamics

The aerodynamics description is the part of the wind turbine that contains the information on how the kinetic energy of the wind is passed on the rotating shaft through the blades. The model of aerodynamic section is based on the assumption of a uniform wind field and thus an equal contribution of energy from each blade. In this case rotor transforms the kinetic energy from the wind passing through the rotor plane to mechanical energy at the shaft ant through the generator to electrical power. The obtained power P_r by rotor is

$$P_r = \frac{1}{2}\rho A_r {v_r}^3 C_p(\lambda,\beta) \tag{4}$$

Where the swept area by the rotor is$A_r = \pi R^2$, ρ is the air density, R is the rotor radius, and C_p is the power efficiency coefficient. In addition to deliver power to the wind turbine, the wind will exert a thrust on the rotor plane that is, a force on the nacelle of rotor in the fore-aft direction. The thrust force F_T is given by

$$F_T = \frac{1}{2}\rho A_r {v_r}^2 C_T(\lambda,\beta) \tag{5}$$

Where C_T is much like C_p and they are both functions of tip speed ratio λ and pitch angleβ, that are shown in Fig. 4. Also tip speed-ratio defined as

$$\lambda = \frac{w_r R}{v_r} \tag{6}$$

Finally the relation between the obtained power by rotor P_r and resulting torque Q_r can be represented as

$$Q_r = \frac{P_r}{w_r}$$
$$Q_r = \frac{1}{w_r}\frac{1}{2}\rho A_r {v_r}^3 C_p(\lambda,\beta) \qquad (7)$$

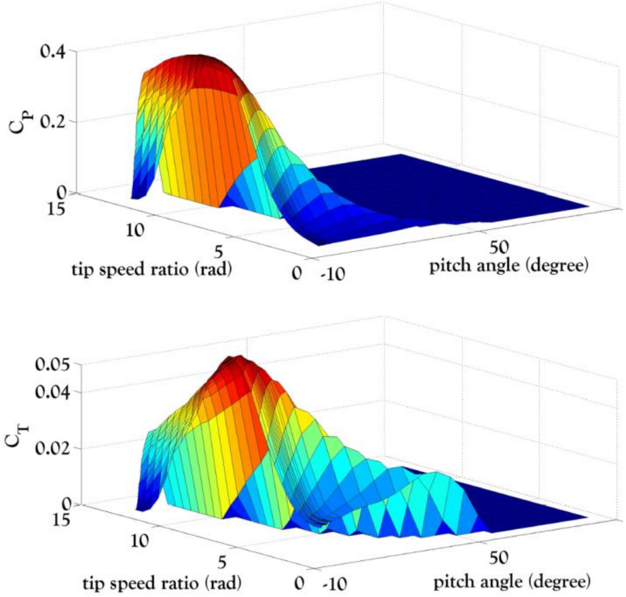


Fig.4. The power coefficient C_p (above) and the thrust coefficient C_T (below) as the functions of the tip speed-ratio λ and pitch angle β

Drive Train

The structural model of drive train is suggested in Fig. 5. The model consists of the following components:

- $\mathrm{I_r}$: The inertia of the rotor and the low-speed shaft
- $\mathrm{I_g}$: The inertia of the gearbox and the high-speed shaft and generator
- $\mathrm{N_g}$: The gearbox ratio
- $\mathrm{K_s}$ and $\mathrm{D_s}$: The spring constant and viscosity of a massless, viscously damped rotational spring respectively
- $\theta_{\mathrm{torsion}}$: The rotational deformation of the low-speed shaft that can be defined as $\theta_{\mathrm{torsion}} = \theta_{\mathrm{r}} - \frac{\theta_{\mathrm{g}}}{\mathrm{N_g}}$

The dynamics of the drive train model can be described by set of differential equations

$$I_r\dot{w}_r = Q_r - K_s\theta_{torsion} - \mathrm{D_s}(\mathrm{w}_r - \frac{w_g}{N_g})$$
$$I_g\dot{w}_g N_g = -Q_g N_g + K_s\theta_{torsion} + \mathrm{D_s}(\mathrm{w}_r - \frac{w_g}{N_g})$$
$$\dot{\theta}_{torsion} = \mathrm{w}_r - \frac{w_g}{N_g} \qquad (8)$$

Where Q_r and Q_g are the torque imposed at the low speed shaft by the rotor and generator torque respectively.

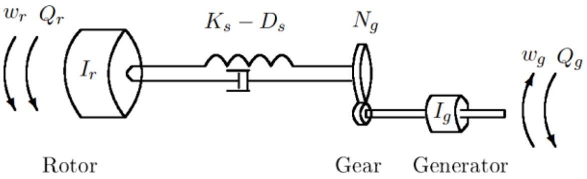


Fig.5. Mechanical equivalent for the drive train

Tower

The tubular steel tower will be deflected in the fore-aft direction due to the thrust force on the rotor. A simple model approximates the deflection with a linear displacement of the nacelle, with the dynamics described by

$$m_t\ddot{z} = -D_t\dot{z} - K_t z + F_t \qquad (9)$$

Where the mass of the tower is m_t, D_t is the tower dampener coefficient and K_t is the tower spring coefficient. Also z, $\dot{z}$ and $\ddot{z}$ denote position, velocity, and acceleration of the nacelle, respectively.

Pitch Actuator

To change the angle of the blade towards the wind in the model, a pitch actuator is used for every blade. The pitch actuators are placed in the hub of the wind turbine and consist of a hydraulic driven servo system. The dynamic of hydraulic servo system can be described by

$$\ddot{\beta} = {w_n}^2\beta_{ref} - 2w_n\xi\dot{\beta} - {w_n}^2\beta \qquad (10)$$

Furthermore a first order model of pitch system with a pure time delay is suggested as follows

$$\dot{\beta} = \frac{1}{\tau_\beta}(\beta_{ref} - \beta) \qquad (11)$$

Practical pitch actuator cannot change the pitch angle instantaneously. In fact aforementioned equation interprets this latency between the pitch angle reference β_{ref} and the actual blade pitch angle β by means of a first order relationship.

Where τ_β is the pure time delay constant, ξ and w_n are damping ratio and undamped frequency of the pitch actuator, respectively.

Generator

The generator used in this HAWT model is an asynchronous generator that exerts the torque on the generator side on the drive train. We will consider the generator as a device that attempts to deliver the electrical power P_e specified by the power reference signalP_{ref}. The power is controlled by adjusting the rotor current, which in turn, governs the amount of torque exerted by the generator to the high-speed shaft. Further, we will assume a loss-less generator, meaning that the electrical power equals the product between the generator speed and the generator torque:

$$P_g = Q_g w_g \tag{12}$$

Practical generators cannot change the torque instantaneously. We will model this latency by a first-order relationship between the requested generator torque and the actual generator torque

$$\dot{Q}_g = \frac{1}{\tau_g}(Q_{g_{ref}} - Q_g) \tag{13}$$

Where τ_g is time delay constant and also the desired torque is given by $Q_{g_{ref}} = \frac{P_{ref}}{w_g}$

The Uncertain Parameters of Pitch Actuator

The considered wind turbine has a hydraulic pitch system which is modeled as a second-order system as described in (8). A drop in the hydraulic pressure affects the dynamics of the pitch system by changing the damping ratio and undamped natural frequency from their nominal values ξ_0 and $w_{n_0}$to their values at low pressure ξ_{lp} and $w_{n_{lp}}$ as described as follows[2]

$$\begin{aligned} {w_n}^2 &= (1-\alpha)\,{w_{n_0}}^2 + \alpha {w_{n_{lp}}}^2 \\ 2w_n\xi &= 2(1-\alpha)\xi_0 w_{n_0} + 2\alpha\,\xi_{lp} w_{n_{lp}} \end{aligned} \tag{14}$$

Low hydraulic pressure is characterized as a gradual fault, since it affects control actions of the turbine. Where $\alpha \in [0\ 1]$ is an indicator function for the fault with $\alpha = 0$ and
$\alpha = 1$, corresponding to normal pressure and low pressure respectively, and $\dot{\alpha} \in [-30/s, 30/s]$.
The nonlinear non-affine state-space dynamic model of the HAWT can be described by the set of following equations [9,10]

$$\left\{ \begin{aligned} \dot{w}_r &= \frac{1}{I_r}(Q_r - K_{\mathrm{s}}\theta_{torsion} - \mathrm{D_s}(\mathrm{w}_r - \frac{w_g}{N_g})) \\ \dot{w}_g &= \frac{1}{I_g N_g}(-Q_g N_g + K_{\mathrm{s}}\theta_{torsion} + \mathrm{D_s}(\mathrm{w}_r - \frac{w_g}{N_g})) \\ \dot{\theta} &= \mathrm{w}_r - \frac{w_g}{N_g} \\ \ddot{z} &= \frac{1}{m_t}(-D_t\dot{z} - K_t z + F_t) \\ \dot{Q}_g &= \frac{1}{\tau_g}(\frac{P_{ref}}{w_g} - Q_g) \\ \dot{\beta} &= \frac{1}{\tau_\beta}(\beta_{ref} - \beta) \\ \ddot{\beta} &= {w_n}^2\beta_{ref} - 2w_n\xi\dot{\beta} - {w_n}^2\beta \\ \ddot{\tilde{v}} &= -\frac{1}{\tau_1\tau_2}\tilde{v} - \frac{\tau_1+\tau_2}{\tau_1\tau_2}\dot{\tilde{v}} + \frac{k}{\tau_1\tau_2}\delta \end{aligned} \right. \tag{15}$$

Where state x, input u and disturbance d vectors of system are defined as

$$\begin{aligned} \underline{x} &= [\mathrm{w}_r \quad w_g \quad \theta \quad z \quad \dot{z} \quad Q_g \quad \beta \quad \dot{\beta} \quad \tilde{v} \quad \dot{\tilde{v}}]^T \\ u &= \begin{bmatrix}\beta_{ref} \\ P_{ref}\end{bmatrix} \qquad d = \begin{bmatrix}\bar{v} \\ \delta\end{bmatrix} \end{aligned} \tag{16}$$

3. PROBLEM STATEMENT

The dynamics of wind turbines are highly nonlinear and non-affine and may also contain uncertain parameters. Generally, in a HAWT the control signals are the pitch angle reference and desired output power and the main objective is the output electrical power. The goal of turbine control is tracking of output power when certain power is applied. Furthermore, control performance of HAWT is highly sensitive to variations of disturbances and system parameters. Noting that these variations are noxious factors in the turbine control system, in this paper an IMRANC power control will be presented to reduce their effects. The major contributions of this study are:

- Successful employment of a proper NARX model on behalf of the original nonlinear plant with neglectable error between outputs of the original system and estimated model
- Successful validation of proposed NARX NN model in comparison with the nonlinear non-affine system
- Successful design of feasible IMRANC power controller based on aforementioned NARX model in the presence of parameters uncertainties as well as disturbances
- Successful development of transient responses and disturbance attenuation of the output power tracking
- Successful robustness of the designed system ensuring that all closed-loop performance specifications are satisfied in the presence of the unavoidable model uncertainty when the parameters in the system dynamic are varied
- Successful design of the controller in the event that the control signals don't exceed the allowable limits for the system even with a wide range of system uncertainties
- Successful responses of electrical power control for various power commands, in order to prove afford tracking of different power reference inputs

4. PROPOSED CONTROL SCHEME

A. MLP Networks and Levenberg-Marquardt BP Algorithm

Neural networks have been applied successfully in the identification and control of dynamic systems. The universal approximation capabilities of the multilayer perceptrons (MLPs) make it a popular choice for modeling nonlinear systems and for implementing general-purpose nonlinear controllers [23-25]. The MLPs have been applied effectively to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as the error back-propagation algorithm. This algorithm is based on the error-correction learning rule. As such, it may be viewed as a generalization of an equally popular adaptive filtering algorithm: the ubiquitous least-mean-square (LMS) algorithm. Basically, error back-propagation learning consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an activity pattern (input vector) is applied to the sensory nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass the synaptic weights of the networks are all fixed. During the backward pass, on the other hand, the synaptic weights are all adjusted in accordance with an error-correction rule. Specifically, the actual response of the network is subtracted from a desired (target) response to produce an error signal. This error signal is then propagated backward through the network. The synaptic weights are adjusted to make the actual response of the network move closer to the desired response in a statistical sense. An architectural scheme for an MLP network having three layers is illustrated in Fig. 6. According to three-layer feedforward network in Fig. 6, the net input to unit *i* in layer *k + 1* is [23].

$$n^{K+1}(i) = \sum_{j=1}^{S_k} w^{K+1}\,(i,j)a^{K}(j) + b^{K+1}(j) \tag{17}$$

The output of unit *i* will be

$$a^{K+1}(i) = f^{K+1}(n^{K+1}(j)) \tag{18}$$

For an *M* layer network the system equations in matrix form are given by

$$\begin{aligned} &\underline{a}^{0} = \underline{p} \\ &\underline{a}^{K+1} = \underline{f}^{K+1}(w^{K+1}\underline{a}^{K} + \underline{b}^{K+1}) \\ &K = 0,1,\dots,M-1 \end{aligned} \tag{19}$$

The task of the network is to learn associations between a specified set of input-output pairs

$$\left\{\left(\underline{p}_1,\underline{t}_1\right),\left(\underline{p}_2,\underline{t}_2\right),\dots,\left(\underline{p}_Q,\underline{t}_Q\right)\right\}$$

The performance index for the network is

$$\begin{aligned} V &= \frac{1}{2}\sum_{q=1}^{Q}\left(\underline{t}_q - \underline{a}_q{}^{M}\right)^{T}\left(\underline{t}_q - \underline{a}_q{}^{M}\right) \\ &= \frac{1}{2}\sum_{q=1}^{Q}\underline{e}_q{}^{T}\underline{e}_q \end{aligned} \tag{20}$$

Where $\underline{a}_q{}^{M}$ is the output of the network when the *qth* input, $\underline{p}_q$, is presented, and $\underline{e}_q = \underline{t}_q - \underline{a}_q{}^{M}$ is the error for the *qth* input. For the standard backpropagation algorithm we use an approximate steepest descent rule. The performance index is approximated by

$$\hat{V} = \frac{1}{2}\underline{e}_q{}^{T}\underline{e}_q \tag{21}$$

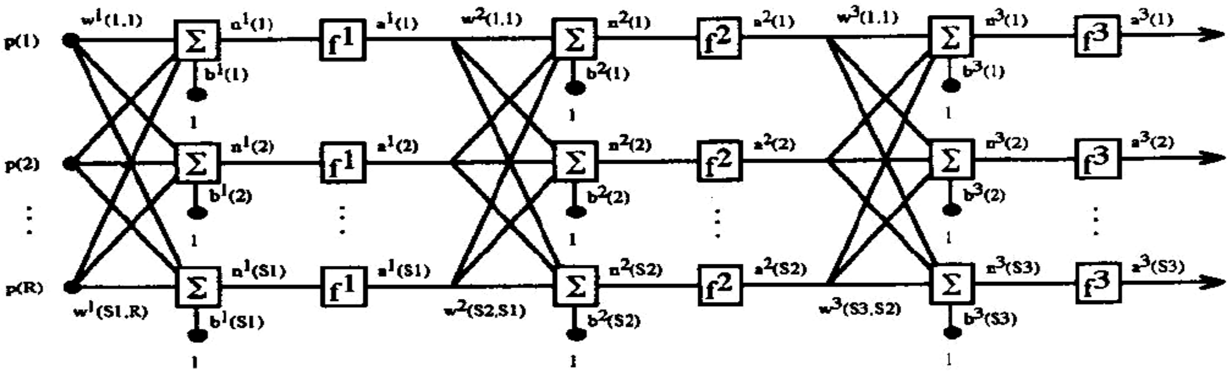


Fig. 6. Structure of an MLP neural network

Where the total sum of squares is replaced by the squared errors for a single input/output pair. The approximate steepest (gradient) descent algorithm is then

$$\Delta w^k(i,j) = -\alpha \frac{\partial \hat{V}}{\partial w^k(i,j)}$$
$$\Delta b^k(i) = -\alpha \frac{\partial \hat{V}}{\partial b^k(i)} \qquad (22)$$

Where α is the learning rate. Define

$$\sigma^k(i) = \frac{\partial \hat{V}}{\partial n^k(i)} \qquad (23)$$

As the sensitivity of the performance index to changes in the net input of unit i in layer k. Now it can be shown, using (l7), (21), and (23), that

$$\frac{\partial \hat{V}}{\partial w^k(i,j)} = \frac{\partial \hat{V}}{\partial n^k(i)} \frac{\partial n^k(i)}{\partial w^k(i,j)} = \sigma^k(i)\, a^{k-1}(j)$$
$$\frac{\partial \hat{V}}{\partial b^k(i)} = \frac{\partial \hat{V}}{\partial n^k(i)} \frac{\partial n^k(i)}{\partial b^k(i)} = \sigma^k(i) \qquad (24)$$

It can also be shown that the sensitivities satisfy the following recurrence relation

$$\underline{\sigma}^k = \dot{F}^k(\underline{n}^k) w^{k+1^T} \underline{\sigma}^{k+1}$$
$$\dot{F}^k(\underline{n}^k) = \begin{bmatrix} \dot{f}^k(n^k(1)) & 0 & \cdots & 0 \\ 0 & \dot{f}^k(n^k(2)) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{f}^k(n^k(S_k)) \end{bmatrix}$$
$$\dot{f}^k(n) = \frac{df^k(n)}{dn} \qquad (25)$$

This recurrence relation is initialized at the final layer

$$\underline{\sigma}^M = -\dot{F}^M(\underline{n}^M)(\underline{t}_q - \underline{a}_q) \qquad (26)$$

The overall learning algorithm now proceeds as follows; first, propagate the input forward using (19); next, propagate the sensitivities back using (25) and (26); and finally, update the weights and offsets using (22) and (24). While backpropagation is a steepest descent algorithm, the Marquardt-Levenberg algorithm [23] is an approximation to Newton's method. Suppose that we have a function $V(\underline{x})$ which we want to minimize with respect to the parameter vector $\underline{x}$, then Newton's method would be

$$\Delta\underline{x} = -[\nabla^2 V(\underline{x})]^{-1} \nabla V(\underline{x})$$

$$V(\underline{x}) = \sum_{i=1}^{N} {e_i}^2(\underline{x}) \qquad (27)$$

Where $\nabla^2 \mathrm{V}(\underline{\mathrm{x}})$ is the Hessian matrix and $\nabla \mathrm{V}(\underline{\mathrm{x}})$ is the gradient. Then it can be shown that

$$\nabla V(\underline{x}) = J^T(\underline{x})\, \underline{e}(\underline{x})$$
$$\nabla^2 V(\underline{x}) = J^T(\underline{x}) J(\underline{x}) + S(\underline{x})$$
$$J(\underline{x}) = \begin{bmatrix} \frac{\partial e_1(\underline{x})}{\partial x_1} & \frac{\partial e_1(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_1(\underline{x})}{\partial x_n} \\ \frac{\partial e_2(\underline{x})}{\partial x_1} & \frac{\partial e_2(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_2(\underline{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\underline{x})}{\partial x_1} & \frac{\partial e_N(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_N(\underline{x})}{\partial x_n} \end{bmatrix}$$
$$S(\underline{x}) = \sum_{i=1}^{N} e_i(\underline{x})\, \nabla^2 e_i(\underline{x}) \qquad (28)$$

Where $\mathrm{J}(\underline{\mathrm{x}})$ is the Jacobian matrix. For the Gauss-Newton method it is assumed that $S(\underline{x}) \approx 0$, and the update (27) becomes

$$\Delta\underline{x} \approx [J^T(\underline{x}) J(\underline{x})]^{-1} J^T(\underline{x}) \underline{e}(\underline{x}) \qquad (29)$$

The Marquardt-Levenberg modification to the Gauss-Newton method is

$$\Delta\underline{x} = [J^T(\underline{x}) J(\underline{x}) + \mu I]^{-1} J^T(\underline{x}) \underline{e}(\underline{x}) \qquad (30)$$

The parameter μ is multiplied by some factor (β) whenever a step would result in an increased $\mathrm{V}(\underline{\mathrm{x}})$. When a step reduces$\mathrm{V}(\underline{\mathrm{x}})$, μ is divided by β. Notice that when μ is large, the algorithm becomes steepest descent (with step $\frac{1}{\mu}$), while for small μ, the algorithm becomes Gauss-Newton. The Marquardt-Levenberg algorithm can be considered as a trust region modification to Gauss-Newton. The key step in this algorithm is the computation of the Jacobian matrix. For the neural network mapping problem, the terms in the Jacobian matrix can be computed by a simple modification to the backpropagation algorithm. The performance index for the mapping problem is given by (20). It is easy to see that this is equivalent in form to (27), where

$$\underline{x} = [w^1(1,1) w^1(1,2) \ldots w^1(S1,R) b^1(1) \ldots b^1(S1) w^2(1,1) \ldots b^M(SM)]^T,$$

and $\mathrm{N} = \mathrm{Q} \times \mathrm{SM}$. Standard backpropagation calculates terms like

$$\frac{\partial \hat{V}}{\partial w^k(i,j)} = \frac{\partial \sum_{m=1}^{SM} e_q{}^2(m)}{\partial w^k(i,j)} \qquad (31)$$

For the elements of the Jacobian matrix that are needed for the Marquardt algorithm we need to calculate terms like

$$\frac{\partial e_q(m)}{\partial w^k(i,j)} \qquad (32)$$

These terms can be calculated using the standard backpropagation algorithm with one modification at the final layer

$$\Delta^M = -\dot{F}^M\left(\underline{n}^M\right) \qquad (33)$$

Note that each column of the matrix in (33) is a sensitivity vector which must be backpropagated through the network to produce one row of the Jacobian. The Levenberg-Marquardt Back Propagation (LMBP) algorithm thus proceeds as follows: (I) Present all inputs to the network and compute the corresponding network outputs (using (19)), and errors $\underline{e}_q = \underline{t}_q - \underline{a}_q{}^M$. Compute the sum of squares of errors over all inputs $V(\underline{x})$ (II) Compute the Jacobian matrix (using (24),(25),(28) and (33)) (III) Solve (30) to obtain $\Delta\underline{x}$ (IV) Recompute the sum of squares of errors using $\underline{x} + \Delta\underline{x}$. If this new sum of squares is smaller than that computed in step I, then reduce μ by β, let $\underline{x} + \Delta\underline{x}$, and go back to step I. If the sum of squares is not reduced, then increase μ by β and go back to step III (V) The algorithm is assumed to have converged when the norm of the gradient in (28) is less than some predetermined value, or when the sum of squares has been reduced to some error goal.

B. IMRNC Design

An IMRANC (Indirect Model Reference Adaptive Neuro Control) is formed by combining an online parameter estimator, which provides estimates of unknown parameters at each instant, with a control law that is motivated from the known parameter case [26], [27]. In the IMRANC, the plant parameters are estimated on-line and used to calculate the controller parameters based on explicit plant model. In IMRANC, a good understanding of the plant and the performance requirements it has to meet allow the designer to come up with a model, referred to as the reference model, that describes the desired Input / Output (I/O) properties of the closed-loop plant. The objective of IMRANC is to find the feedback control law that changes the structure and dynamics of the plant so that its I/O properties are exactly the same as those of the reference model. The typical structure of an IMRANC scheme is shown in Fig.7.

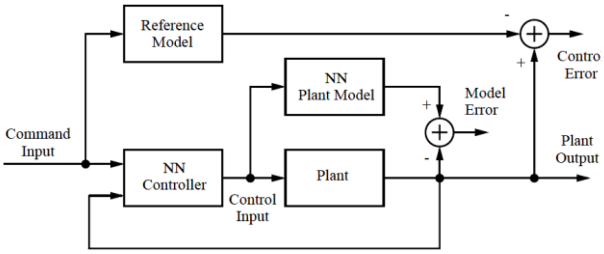


Fig. 7. Typical structure of IMRANC

The nonlinear autoregressive network with exogenous inputs (NARX) network is a recurrent dynamic network, with feedback connections enclosing several layers of the network. The NARX model is based on the linear ARX model, which is commonly used in time-series modeling [20], [21]. The defining equation for the NARX model is

$$y(t) = f(y(t-1), y(t-2), \dots, y\left(t-n_y\right), u(t-1), u(t-2), \dots, y(t-n_u)) \qquad (34)$$

Where, the next value of the dependent output signal $y(t)$ is regressed on previous values of the output signal and previous values of an independent (exogenous) input signal. The NARX networks can learn to predict one time series given past values of the same time series, the feedback input, and another time series, called the external or exogenous time series. Some considerable qualities about NARX networks with gradient-descending learning gradient algorithm can be explained as:

(1)Learning is more efficient in NARX networks than in other neural network (the gradient descent is better in NARX)

(2)These networks converge much faster and generalize better than other networks

In this paper an NARX network is utilized as a model of the plant that you want to control. The NARX model can be trained using the training functions described in MLP Networks and LMBP algorithm (referring to section III). The diagram of the proposed NARX network for plant model having one hidden layer is illustrated in Fig. 8.

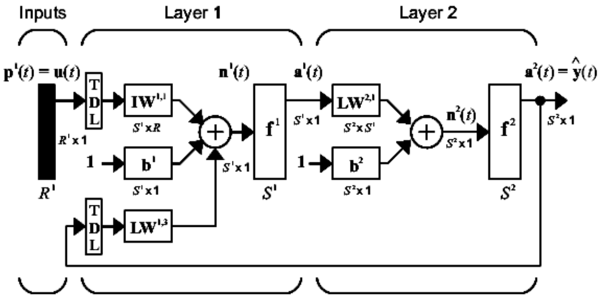


Fig. 8. NARX network for plant model

Where the TDL (Tapped Delay Line) components are input and feedback delays, S^1 and S^2 are number of neurons in the hidden and output layers respectively, u(t) is network input, $\hat{y}(t)$ is network target, f^{1} and f^{2} are transfer functions in hidden and output layers respectively and R^1 is number of elements in input vector. The NARX model for approximation of plant can be applied in various ways, but it seems to be simpler by using a feedforward neural network with the embedded memory (the first TDL), plus a certain delay from the output of the second layer to input (the second TDL). The IMRANC architecture has two subnetworks. One subnetwork is the model of the plant that you want to control and the other subnetwork is the controller (according to Fig. 7). At first the I/O data is collected from the implemented HAWT and create and train aforementioned NARX network based on LMBP algorithm. Where the implemented HAWT based on block diagram of the full model (in Fig.2) and the nonlinear non-affine state-space dynamic model (15)-(16) is presented in Fig. 15 (*Appendix I*). Afterward total IMRANC system is designed and the NARX model is inserted inside. In this way a feedforward network is selected for IMRANC system and the feedback connections are then added. Also, learning in the plant model subnetwork is turned off, since it has already been trained and the controller subnetwork is only trained. The final IMRANC network can be viewed in Fig.9.

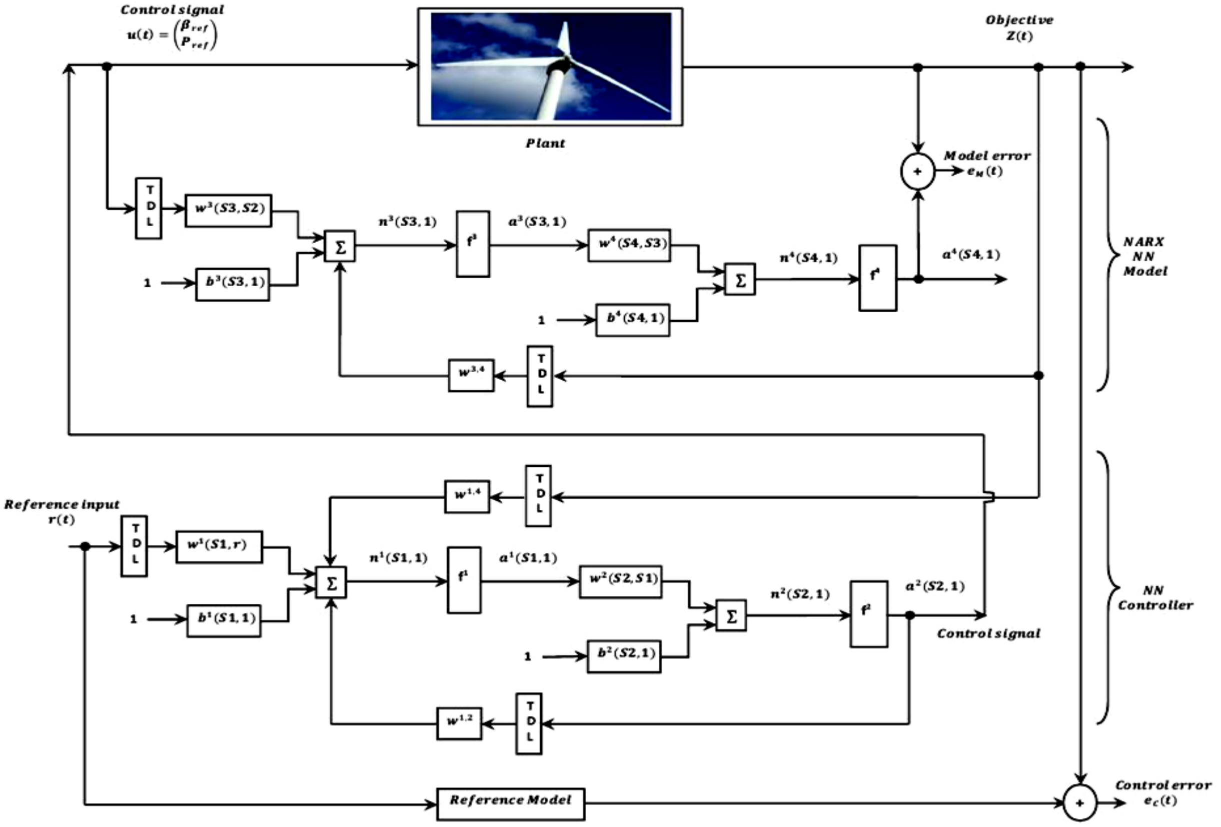


Fig.9. The Proposed Indirect Model Reference Adaptive Neuro Control System (IMRANC)

According to Fig. 9 layers 3 and 4 make up the NARX neural network subnetwork of the plant model and layers 1 and 2 make up the neural network controller. In this diagram f^{1} and f^{2} are transfer functions in hidden and output layers of NN controller respectively and also the transfer functions in hidden and output layers of NARX NN plant model are symbolized by f^{3} and f^{4} respectively. The tangent-sigmoid (TanSig) transfer function was used for hidden layers of controller as a nonlinear function, and model and the linear (Lin) transfer function is also utilized for output layers of them. The mathematically equation of TanSig transfer function is given as

$$a = TanSig(n)$$

$$TanSig(n) = \frac{2}{1+e^{-2n}} - 1 \qquad (35)$$

In other words f^1 and f^3 are tangent-sigmoid and also f^2 and f^4 are linear transfer functions. As a result, each two-layer subnetwork of NN model and NN controller are designed by TanSig-Lin architecture. Where

TanSig output neurons are often used for pattern recognition problems, while linear output neurons are used for function fitting problems.

In this research both of NARX NN model and NN controller have only one hidden layer and number of neurons in their hidden layer are 15 (S1=S3=15). The number of output neurons in NN model and NN controller are 1 and 2 respectively, because HAWT has one objective z(t) and control signals u(t). Where desired output power P_{ref} and pitch angle reference β_{ref} are defined as the control signals and electrical power P_e of HAWT is described as the main control objective that the P_e equals the product between the generator speed w_g and the generator torque Q_g :

$$P_e = w_g Q_g \qquad (36)$$

The first and second samples of objective z(t), control signals u(t) and reference input r(t) as lags of the input and output of the system are given for two-step ahead prediction in IMRANC system. In other words the TDL of input and feedbacks are selected 1:2.

5. SIMULATION RESULTS

In this section, we show the effectiveness of the proposed method by performing some simulation studies over a 4.8 MW horizontal axis wind turbine. The parameters of this turbine are presented in Table I (*Appendix II*) [2].

The nonlinear non-affine state-space dynamic of the HAWT described in (15) leaves a dynamic system with four inputs: the mean wind speed $\bar{v}$ and the white noise δ as the disturbances and on the other hand the pitch angle reference signal β_{ref} and power reference P_{ref} as the control signals. The pitch angle reference and power reference are considered controllable inputs and the wind speed and white noise are uncontrollable disturbances. Furthermore the produced power P_e is defined as the main control objective. The proposed IMRANC uses two neural networks: a controller network and a plant model network, as shown in the Fig.7. The NARX plant model is identified first and we will approve that the proposed NARX NN model can estimate states of the original plant (5) without any model error between the plant and NARX model.

Fig.10 compares states of the proposed NARX NN model in comparison with the nonlinear non-affine system (15), where the dashed lines denote the estimated state variables $\underline{\hat{x}}(t)$ by the NN model and solid lines indicate states of the original plant $\underline{x}(t)$. Accordingly, design of a NARX NN model is required to satisfy $\underline{\hat{x}}(t) \rightarrow \underline{x}(t)$ and this condition guarantees that the steady-state errors between $\underline{\hat{x}}(t)$ and $\underline{x}(t)$ converges to 0. As it is evident in this figure, the proposed model estimates states of the nonlinear plant (15) without any steady-state error, which means that the NARX model can represent the original system in the pre-specified domains with a suitable approximation.

The controller is then trained so that the plant output follows the reference model output. In this part we will study the produced electrical power tracking performance in the presence of parameter variations in the pitch actuator component and also both of the disturbances using the proposed IMRANC scheme (Fig.9) based on LMBB algorithm.

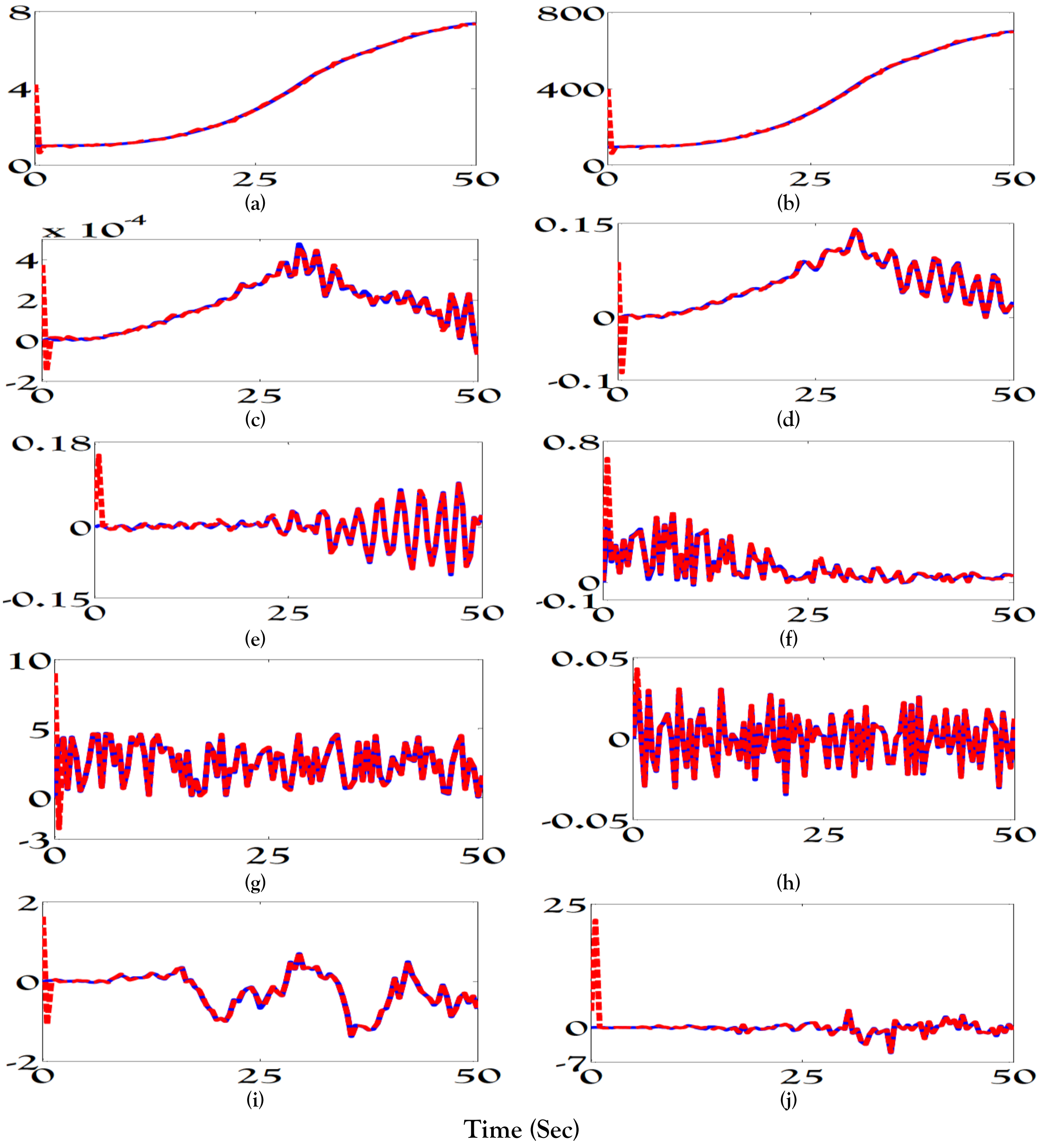
Fig. 11 shows the mean wind speed $\bar{v}$, wind speed seen by the rotor plane v_r and thrust force on the hub of tower F_t . As can be observed, the behaviors of $\bar{v}$ and v_r can changes the thrust force of the rotor. The increasing of the $\bar{v}$ and v_r increases F_t until at t=30 sec, the highest value of thrust force is occurred ($3.3x10^5$ N) and when $\bar{v}$ is reached to constant value 30 m/s, $\mathrm{F_t}$ is decreased to zero again.

Fig. 12 (a) illustrates electrical power P_e regulation responses of HAWT at different output power commands (P_{er} (MW)) by using the proposed IMRANC. Furthermore the pitch angle referenceβ_{ref} and the desired output power P_{ref} as two control signals for various power commands are shown in Figs 12 (a) and (b) respectively. According to Fig. 12 (a), the proposed system has satisfactory performance for various power commands in order to generate different produced output power.

Fig. 13 shows tracking of output power and related control signals based on a rectangular trajectory. From the mentioned figure, you can see that the plant model output does follow the reference trajectory with the correct critically-damped response and without the steady state error, even though the input sequence was not the same as the input sequence in the training data. In addition referring to figs. 13 (b) and (c) the pitch angle reference and desired output power as the input signals are limited to [0 , 4.8] (MW) and [-2 , 40] (°) respectively. Also fig. 13 validates the effectiveness of the proposed controller like robustness and good load disturbance attenuation and accurate tracking, even in the presence of parameter variations (changing the damping ratio and undamped natural frequency from their nominal) due to a drop in the hydraulic pressure and also disturbances.

For assess the effectiveness and the robustness of the proposed method, two sets of HAWT system are defined. These sets are symbolized by $\mathrm{Sys_{nom}}$ and $\mathrm{Sys_{low}}$, in the event that the set $\mathrm{Sys_{nom}}$ is indeed the nominal model (when hydraulic pressure is normal, $\alpha = 0$) and the set $\mathrm{Sys_{low}}$ introduce variations in the physical parameters (when hydraulic pressure is low, $\alpha = 1$).

Fig. 14 shows tracking of output power to a certain command and control signals against parameter uncertainties, with regard to the proposed method. These figures illustrate the input signals and power responses for sets $\mathrm{Sys}_{\mathrm{nom}}$ and $\mathrm{Sys}_{\mathrm{low}}$, when the damping ratio and undamped natural frequency parameters in the hydraulic pitch system are varied based on α. In one world when hydraulic pressure is normal the parameters of the pitch actuator section are $\alpha = 0, \xi_0 = 0.6\frac{rad}{s} and\ w_{n_0} = 11.11\ \frac{rad}{s}$ and when hydraulic pressure is low and a fault is occurred, the parameters of the pitch actuator section are transformed to $\alpha = 1, \xi_0 = 0.9\ \frac{rad}{s} and\ w_{n_0} = 3.42\ \frac{rad}{s}$ .It is clear that the system has good robustness when the parameters in the system dynamics are varied in the aforementioned ranges.



(a) (b)

(c) (d)

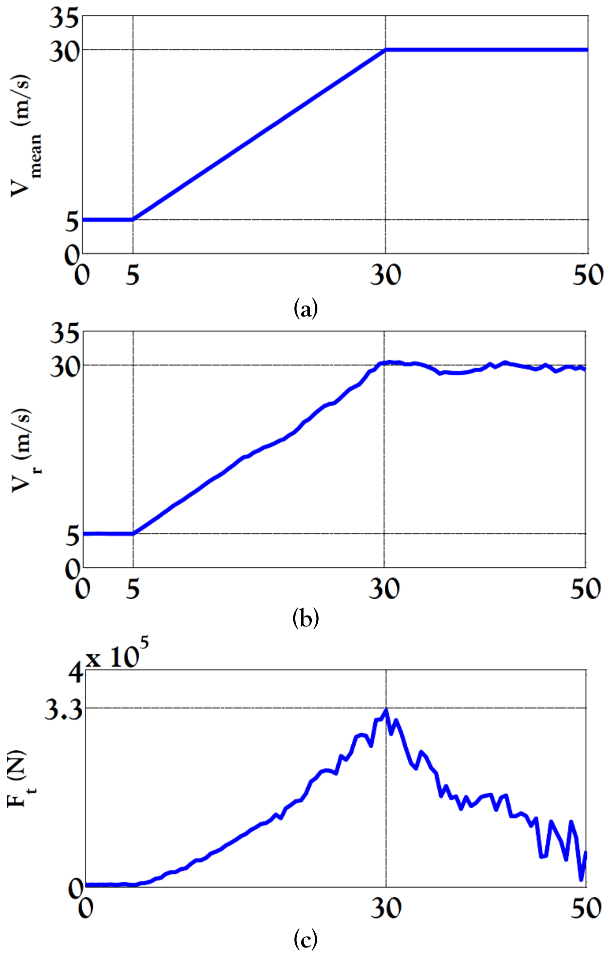
(e) (f)

(g) (h)

(i) (j)

Fig.10. States of the proposed NARX NN model $\underline{\hat{x}}(t)$ (dashed) and the original plant $\underline{x}(t)$ (solid)

(a) Angular speed of rotor shaft w_r (rad/s)
(b) Angular speed of generator shaft w_g (rad/s)
(c) Rotational torsion angle of the low-speed shaft θ (°)
(d) Position of top of tower z (m)
(e) Velocity of top of tower $\dot{z}$ (m/s)

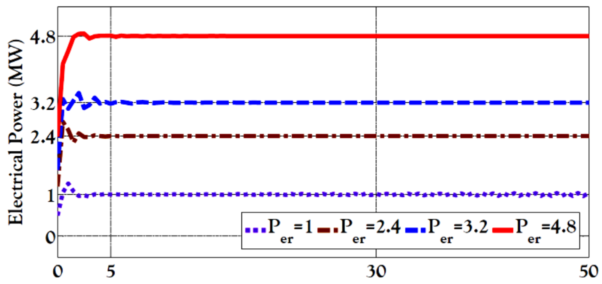


(a)

(b)

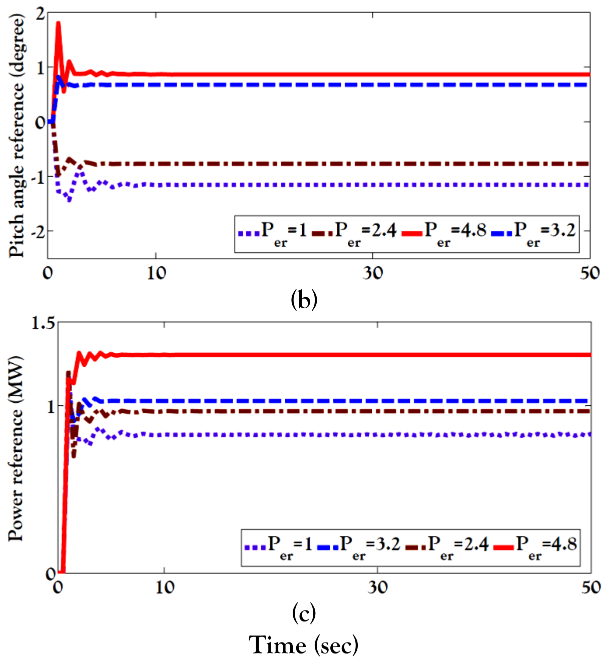
(c)

Fig.11. (a) mean wind speed as the disturbance (b) wind speed seen by the rotor plane (c) thrust force on the hub of tower



(a)

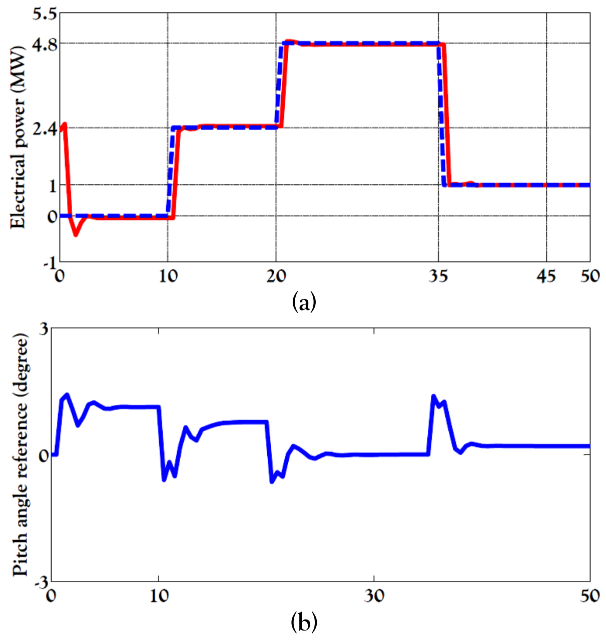
(f) Generator shaft torque Q_g (Nm)
(g) Blade pitch angle β (°)
(h) Pitch angle velocity $\dot{\beta}$ $(°/s)$
(i) Turbulent wind speed $\tilde{v}$ (m/s)
(j) Turbulent wind acceleration $\tilde{\dot{v}}$ (m/s^2)



(b)

(c)

Fig.12. (a) Electrical power P_e (b) Pitch angle reference β_{ref} (c) Desired output power P_{ref}



(a)

(b)

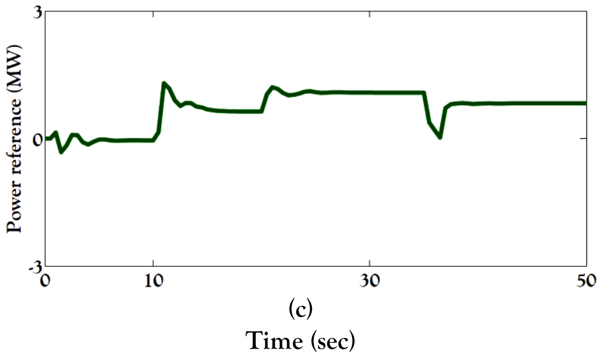


Fig.13. (a) Electrical power tracking response (solid) and reference trajectory (dashed) (b) Pitch angle reference as the first signal control (c) Desired output power as the second signal control

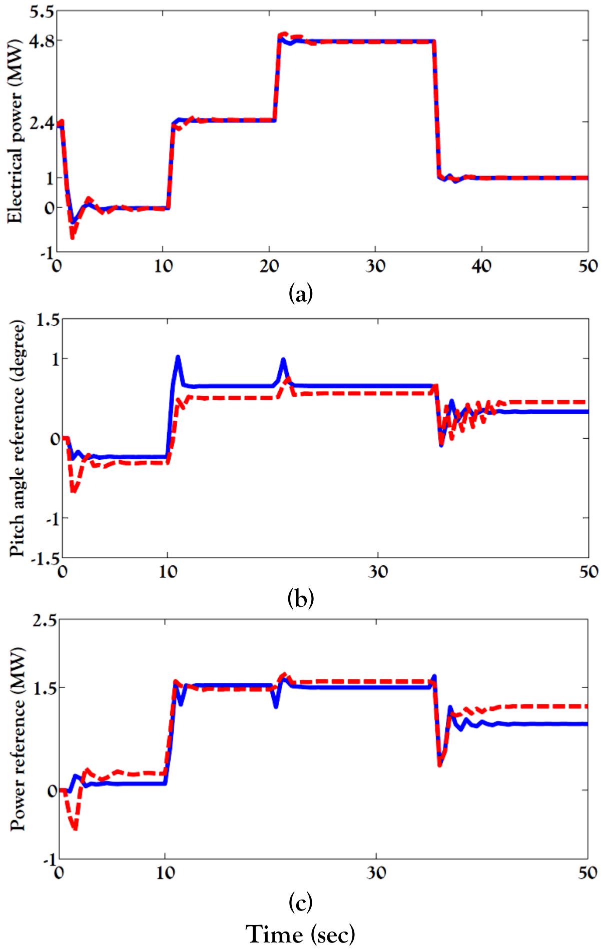


Fig.14. The responses of Sys_{nom} (solid) and Sys_{low} (dashed) with varying ξ and w_n (a) Electrical power (b) Pitch angle reference (c) Desired output power

6. CONCLUSION

In this paper, an IMRANC has been designed for output electrical power tracking and disturbance attenuation of an HAWT. First to identify real system, the NARX neural network model was employed. Afterward, we have shown that the proposed model can accurately represent the original system without steady state model error. The IMRANC scheme is then designed based on the NARX model and trained using LMBB algorithm. The simulation results on the turbine were shown that the proposed control approach has robustness, precise tracking and good disturbances attenuation against disturbances and parameter variations. The major achievements of this research are: (i) The proposed NARX model accurately represent the original nonlinear system (ii) the successful features of both the robustness and nominal performance have been presented by simulation records (iii) The performance requirements like good disturbance rejection, tracking and fast transient responses in the proposed method were successfully assess (iv) The control signals did not exceed the allowable limit (0 to 4.8 MW for desired output power and -2 to 40 degree for pitch angle reference) for two defined sets even with a wide range of system uncertainties.

REFERENCES

[1] A. S. Yilmaz, Z. Ozer, **“Pitch angle control in wind turbines above the rated wind speed by multi-layer perceptron and radial basis function neural networks”**, *Expert Systems with Applications*, Vol. 36, No. 6, ,pp. 9767–9775, 2009.

[2] C. Sloth, T. Esbensen, and J. Stoustrup, **“Robust and fault-tolerant linear parameter-varying control of wind turbines”**, *Mechatronics*, Vol. 21, No. 4, 2011.

[3] H. Camblong, **“Digital robust control of a variable speed pitch regulated wind turbine for above rated wind speeds”**, *Control Engineering Practice*, Vol. 16, No. 8, pp. 946–958, 2008.

[4] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, **“Fault Tolerant Control of Wind Turbines – a benchmark model”**, *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2009.

[5] O. Ognyanova, P. Petkov, E. Haralanova, and T. Puleva, **“Robust control of a wind turbine”**, *7th Mediterranean Conference and Exhibition on Power Generation, Transmission*, Distribution and Energy Conversion, 2010.

[6] K. E. Johnson, L. Y. Pao, M. J. Balas, and L. J. Fingersh, **“Control of variable-speed wind turbines”**, *IEEE Control Systems Magazine*, pp. 70-81, 2006.

[7] T. Esbensen, C. Sloth, **“Fault diagnosis and fault-tolerant control of wind turbines”**, *Thesis*, 2009.

[8] A. Perdana, **“Dynamic models of wind turbines”**, *PHD Thesis*, 2008.

[9] C. Dobrila, R. Stefansen, **“Fault tolerant wind turbine control”**, *MSc Thesis*, 2007.

[10] K. Hammerum, **“A fatigue approach to wind turbine control”**, *PHD Thesis*, 2006.

[11] V. Azimi, M. A. Nekoui, and A. Fakharian., **“Robust multi-objective H_2/ H_∞ tracking control based on T-S fuzzy model for a class of nonlinear uncertain drive systems”**, *Proceeding of the Institution of Mech. Eng. Part I-Journal of Systems and Control Engineering*, Vol. 226, No. 8, , pp. 1107-1118, 2012.

[12] V. Azimi, A. Fakharian, and M. B. Menhaj **“Position and current control of an Permanent-Magnet Synchronous Motor by using loop-shaping methodology: blending of H_∞ mixed-sensitivity problem and T-S fuzzy model scheme”**, *Journal of Dynamic Systems Measurement and Control-Transactions of the ASME*, Vol.135, No. 5, pp. 051006-1–051006-11, 2013.

[13] V. Azimi, M. B. Menhaj, and A. Fakharian, **“Robust H2/ H∞ Control for a Robot Manipulator Fuzzy System”**, *13th Iranian Conference on Fuzzy Systems (IFSC)*, Iran, 2013.

[14] V. Azimi, M. B. Menhaj, and A. Fakharian, **“Fuzzy Robust Control of MIMO Nonlinear Uncertain systems”**, *13th Iranian Conference on Fuzzy Systems (IFSC)*, Iran, 2013.

[15] V. Azimi, A. Fakharian, and M. B. Menhaj, **“Robust Mixed-Sensitivity Gain-Scheduled H∞ tracking control of a nonlinear Time-Varying IPMSM via a T-S fuzzy model”**, *9th France-Japan & 7th Europe-Asia Congress on and Research and Education in Mechatronics (REM)*, France, 2012.

[16] A. Fakharian, V. Azimi, **“Robust mixed-sensitivity H∞ control for a class of MIMO uncertain nonlinear IPM synchronous motor via T-S fuzzy model”**, *Methods and Models in Automation and Robotics (MMAR)*, Poland, 2012.

[17] V. Azimi, M. A. Nekoui, and A. Fakharian, “**Speed and torque control of induction motor by using robust H∞ mixed-sensitivity problem via T-S fuzzy model”**, *Iranian Conference on Electrical Engineering (ICEE)*, Iran, 2012.

[18] V. Azimi, M. B. Menhaj, and A. Fakharian, **“Fuzzy Mixed-Sensitivity Control of Uncertain Nonlinear Induction Motor”**, *Majlesi Journal of Electrical Engineering*, Vol. 8, No. 2, pp. 45-53, 2014.

[19] V. Azimi, M. B. Menhaj, and A. Fakharian, **“Tool position tracking control of a nonlinear uncertain flexible robot manipulator by using robust H_2 /H∞ controller via T–S fuzzy model”**, *Sadhana*, Vol. 40, No. 2, pp. 307-333, 2015.

[20] A. Dzielinski, **“Neural network based NARX models in nonlinear adaptive control”**, *Int. J. Appl. Math. Comput. Sci.*, Vol.12, No.2, pp. 235-240, 2002.

[21] H. T. Siegelmann, B. G. Horne, and C. L. Giles, **“Computational capabilities of recurrent NARX neural networks”**, *IEEE Transactions on systems, man*, and cybernetics-Part B: Cybernetics, Vol. 27, No. 2, 1997.

[22] E. Diaconescu, **“The use of NARX Neural Networks to predict Chaotic Time Series”**, *Wseas Transactions on computer research*, Vol. 3, No. 3, pp. 182-191, 2008.

[23] M. T. Hagan, M. B. Menhaj, **“Training feedforward networks with the Marquardt algorithm”**, *IEEE Transactions on neural networks*, Vol. 5, No. 6, 1994.

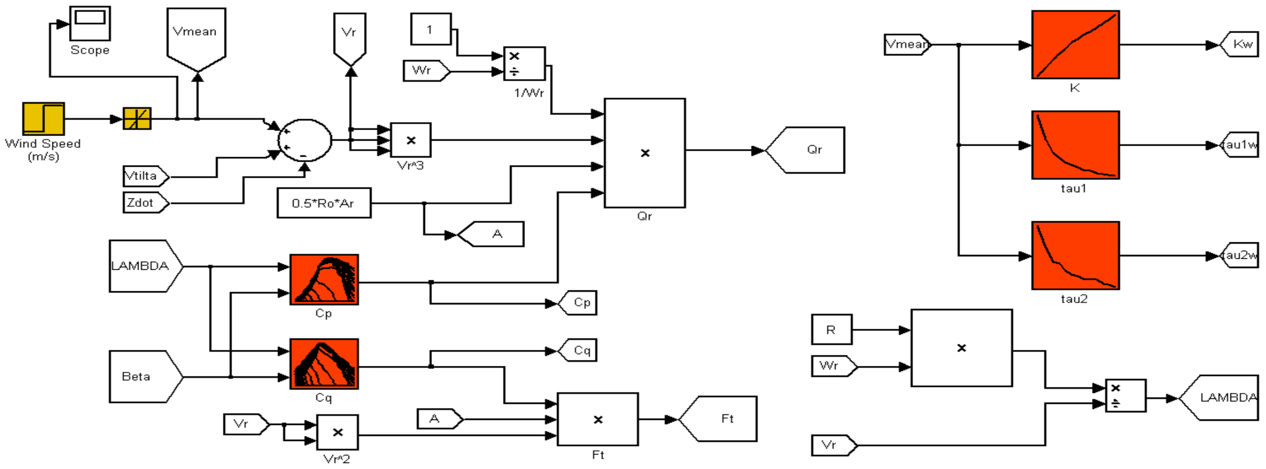
[24] S. Suresh, S. N. Omkar, and V. Mani, **“Parallel implementation of back-propagation algorithm in networks of workstations”**, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, No. 1, pp. 24-34, 2005.

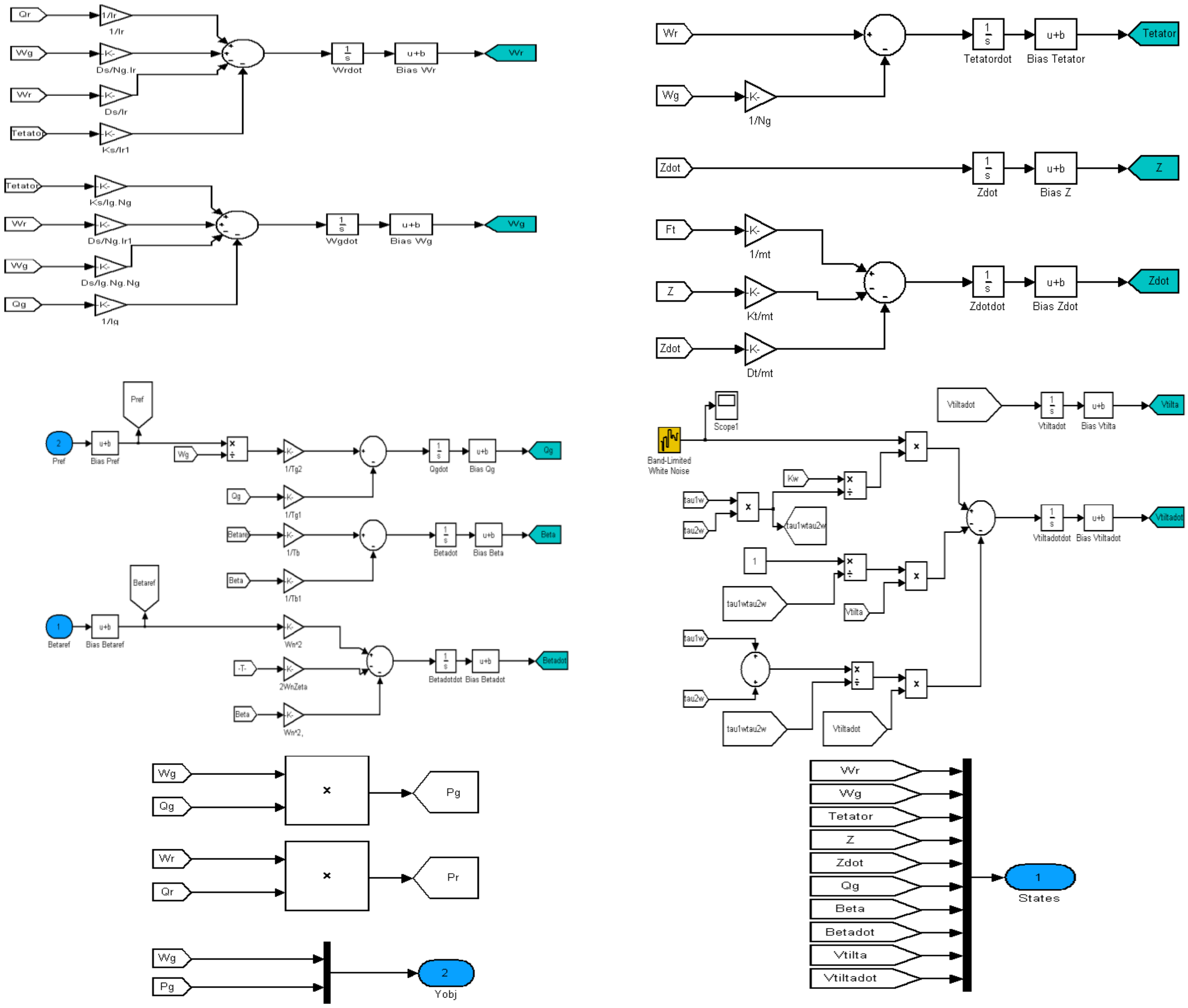
[25] J. J. Rubio, **“Modified optimal control with a back propagation network for robotic arms”**, *IET control theory and applications* , Vol. 6, No. 14, pp. 2216-2225, 2012.

[26] W. Yu, A. S. Poznyak, **“Indirect adaptive control via parallel dynamic neural networks”**, *IEE Proc.-Control Theory Appl.*, Vol. 146, No. 1, 1999.

[27] Q. I. Ruiyun, A. B. Mietek, **“Indirect adaptive controller based on a self-structuring fuzzy system for nonlinear modeling and control”**, *Int. J. Appl. Math. Comput. Sci.*, Vol. 19, No. 4, pp. 619-630, 2009.

Appendix I. The Implemented HAWT





Appendix II. Nominal Parameter Values Of HAWT

Parameter	Value	Description
R (m)	57.5	Rotor radius
ρ (kg/m^3)	1.225	Air density
A_r(m^2)	10387	Swept area by the rotor
I_r (kg m^2)	55.10^6	Inertia of the rotor and low-speed shaft
I_g(kg m^2)	390	Inertia of the gearbox and high-speed shaft and generator
N_g	95	Gearbox ratio
$K_S\ (GNm/rad)$	2.7	Spring constant of rotational spring
$D_S\ (KNm/(rad/s))$	945	Viscosity of rotational spring
$m_t\ (t)$	484	Mass of the tower
$D_t(N/(m/s))$	66.7	Tower dampener coefficient
$K_t(MN/m)$	2.55	Tower spring coefficient
$\xi_0\ (rad/s)$	0.6	Damping ratio of the pitch actuator in normal pressure
$w_{n0}\ (rad/s)$	11.11	Undamped frequency of the pitch actuator in normal pressure
$\xi_{lp}\ (rad/s)$	0.9	Damping ratio of the pitch actuator in low pressure

$w_{nlp}\ (rad/s)$	3.42	Undamped frequency of the pitch actuator in low pressure
$\tau_{\beta}(ms)$	10	Pure time delay constant in pitch actuator
$\tau_{g}(ms)$	10	Pure time delay constant in generator
η_{g}	0.92	Generator efficiency