

# Smart Algorithms to Control a Variable Speed Wind Turbine

Nabil Farhane<sup>1</sup>, Ismail Boumhidi<sup>1</sup>, Jaouad Boumhidi<sup>2</sup> \*

<sup>1</sup> LESSI Laboratory, Department of Physics, Faculty of Sciences, Sidi Mohammed Ben Abdellah, Fez, (Morocco)

<sup>2</sup> LIIAN Laboratory, Department of Computer Science, Faculty of Sciences Sidi Mohammed Ben Abdellah, University of Fez (Morocco)

Received 2 June 2017 | Accepted 15 July 2017 | Published 21 August 2017



## ABSTRACT

In this paper, a robust adaptive fuzzy neural network sliding mode (AFNNSM) control design is proposed to maximize the captured energy for a variable speed wind turbine and to minimize the efforts of the drive shaft. Fuzzy neural network (FNN) is used to improve the mathematical system model, by the prediction of model unknown function, which is used by the Sliding mode control approach (SMC) and enables a lower switching gain to be used despite the presence of large uncertainties. As a result, the used robust control action did not exhibit any chattering behavior. This FNN is trained on-line using the backpropagation algorithm (BP). The particle swarm optimization (PSO) algorithm is used in this study to optimize the learning rate of BP algorithm in order to improve the network performance in term of the speed of convergence. The stability is shown by the Lyapunov theory and the trajectory tracking errors converge to zero without any oscillatory behavior. Simulations illustrate the effectiveness of the designed method.

## KEYWORDS

Adaptive Fuzzy Neural Network Sliding Mode (AFNNSM), Sliding Mode Control (SMC), Particle Swarm Optimization (PSO), Variable Speed Wind Turbine.

DOI: 10.9781/ijimai.2017.08.001

## I. INTRODUCTION

**W**IND energy is an abundant renewable source of new electrical generation capacity in the world, and it is exploited by converting the kinetic energy of moving air mass into electricity; therefore, it is necessary to introduce tools to make these installations more profitable [1]. Practically, there are two main types of horizontal axis wind turbines: fixed speed and variable speed [2]. In this study, we consider the case of variable speed, due to its great ability in the extraction of energy. In addition to that, variable speed system is more complex and requires an efficient control strategy [3]. Several studies have been devoted to the control of the aeroturbine mechanical as well as the electrical components. This work is devoted to the mechanical part (aeroturbine), with the main objective of designing a controller in order to maximize the energy captured from the wind and minimize the stress on the drive train shafts that takes into consideration the nonlinear nature of the system behavior and the flexibility of the drive-train shaft. The proposed control structure also overcomes the drawbacks of some existing control methods.

The sliding mode control [4-7], is widely used in the control of a variable speed wind turbine. This is due to its property of robustness with respect to uncertainties and disturbances [8],[9]. The chattering behavior is especially the main problem in the design of SMC.

One possible method to solve this problem is the boundary layer approach [6]. This technique has given good results when the system uncertainties are small. However, when these uncertainties are large, a high gain is needed and higher amplitude of chattering is produced. In order to solve this problem, we proceed to use FNN for the estimation of the unknown model function so that the system uncertainties can be kept small and hence enable a lower switching gain to be used. The designed method is a combination of traditional SMC and FNN with online adaptation of the parameters.

Intelligent systems such as fuzzy systems and neural networks are being used successfully in an increasing number of application areas [10]-[14]-[23]. Fuzzy neural networks (FNNs) have the low-level learning and computation power of neural networks, and the high-level human-like thinking and reasoning of fuzzy theory. The proposed control consists of the so called equivalent control added with robust control term, the FNN predicted unknown terms are incorporated in the equivalent control component, thus enable the robust component to be used with a small gain which is responsible to compensate the network errors prediction. The FNN is trained on-line using the backpropagation algorithm (BP). The learning rate is one of the parameters of BP algorithm which have a significant influence on results; learning rate which is too small or too large may not be favourable for convergence. Particle swarm optimization (PSO) algorithm is used in this work to optimize this parameter in order to improve the training speed.

This study is organized as follows. The next section presents the wind turbine modelling. Section 3 shows the design of the proposed adaptive fuzzy neural network sliding mode control. In Section 4, simulation results are provided to demonstrate the robust control performance of the proposed (AFNNSM) control. Finally, in section 5 a concluding remark is given.

\* Corresponding author.

E-mail addresses: [nabilfarhane84@gmail.com](mailto:nabilfarhane84@gmail.com) (Nabil Farhane), [iboumhidi@hotmail.com](mailto:iboumhidi@hotmail.com) (Ismail Boumhidi), [jaouad.boumhidi@usmba.ac.ma](mailto:jaouad.boumhidi@usmba.ac.ma) (Jaouad Boumhidi).

## II. WIND TURBINE MODELIZATION

Wind energy across a surface  $A_v$  depends on the cube of the wind speed  $v$ , and the density of the air  $\rho$ . This energy is given by:

$$P_v = \frac{1}{2} A_v \rho v^3 \quad (1)$$

where:

$$A_v = \pi R^2 \quad (2)$$

$R$  is the radius of the rotor. A variable speed wind turbine is composed of aeroturbine, a gearbox, and a generator. The aerodynamic power captured by the rotor is given as follows:

$$P_a = \frac{1}{2} \rho \pi R^2 C_p(\lambda, \beta) v^3 \quad (3)$$

Where  $C_p$  is the power coefficient,  $\beta$  is the pitch angle and the tip-speed ratio,  $\lambda$ , is given as follows:

$$\lambda = \frac{\omega_r R}{v} \quad (4)$$

With  $\omega_r$  is the rotor speed. By using the relationship:

$$P_a = \omega_r T_a \quad (5)$$

The aerodynamic torque expression is:

$$T_a = \frac{1}{2} \rho \pi R^3 C_q(\lambda, \beta) v^2 \quad (6)$$

Where:

$$C_q(\lambda, \beta) = \frac{C_p(\lambda, \beta)}{\lambda} \quad (7)$$

$C_q(\lambda, \beta)$  is the torque coefficient.

In the literature, the wind turbine is presented by the modelling of the mechanical part [15]-[17], or by that of the electrical part [18], [19]. In this paper, we are interested in the modelling of the mechanical part of the variable speed wind turbine, which is presented by a two-mass model that is shown in Fig. 1.

The dynamics of the rotor is characterized by a differential equation of the first order:

$$J_r \dot{\omega}_r = T_a - T_{ls} - K_r \omega_r \quad (8)$$

$J_r$  and  $K_r$  are respectively the rotor inertia and the rotor friction coefficient.

The low-speed shaft torque  $T_{ls}$  resulting effects of friction and torque generated by the differences between the rotor angular velocity  $\omega_r$  and that of the output shaft  $\omega_{ls}$  (see Fig. 1):

$$T_{ls} = B_{ls}(\theta_t - \theta_{ls}) + K_{ls}(\omega_r - \omega_{ls}) \quad (9)$$

$B_{ls}$  and  $K_{ls}$  are respectively the shaft stiffness coefficient and the shaft damping coefficient.

$\theta_r$  and  $\theta_{ls}$  are respectively the rotor side angular deviation and the gearbox side angular deviation.

The relationship between the high-speed shaft torque  $T_{hs}$  and the generator electromagnetic torque  $T_{em}$  is given by:

$$J_g \dot{\omega}_g = T_{hs} - K_g \omega_g - T_{em} \quad (10)$$

$J_g$ ,  $\omega_g$  and  $K_g$  are respectively the generator inertia, the generator speed and the generator friction coefficient. Let's assume an ideal gearbox with transmission ratio  $n_g$ , we get:

$$n_g = \frac{T_{ls}}{T_{hs}} = \frac{\omega_g}{\omega_{ls}} = \frac{\theta_g}{\theta_{ls}} \quad (11)$$

$\theta_g$  is the generator side angular deviation.

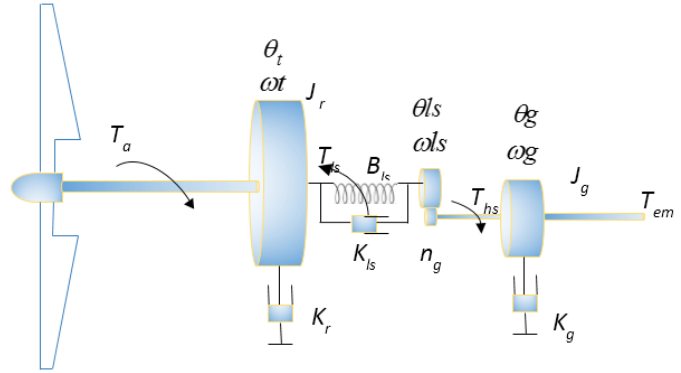


Fig. 1. Two-mass model of wind turbine.

Replacing the time derivative of  $T_{ls}$  from (9) and using (10) and (11) we get the following dynamic system model (12):

$$\begin{bmatrix} \dot{\omega}_r \\ \dot{\omega}_g \\ \dot{T}_{ls} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \omega_t \\ \omega_g \\ T_{ls} \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{bmatrix} T_a + \begin{bmatrix} b_{12} \\ b_{22} \\ b_{32} \end{bmatrix} T_{em}$$

Where:

$$\begin{aligned} a_{11} &= -\frac{K_r}{J_r}, & a_{12} &= 0, & a_{13} &= -\frac{1}{J_r}, & a_{21} &= 0, \\ a_{22} &= -\frac{K_g}{J_g}, & a_{23} &= \frac{1}{n_g J_g}, & a_{31} &= \left( B_{ls} - \frac{K_{ls} K_r}{J_r} \right), \\ a_{32} &= \frac{1}{n_g} \left( \frac{K_{ls} K_g}{J_g} - B_{ls} \right), & a_{33} &= -K_{ls} \left( \frac{J_r + n_g^2 J_g}{n_g^2 J_g J_r} \right), \\ b_{11} &= \frac{1}{J_r}, & b_{12} &= 0, & b_{21} &= 0, & b_{22} &= -\frac{1}{J_g}, & b_{31} &= \frac{K_{ls}}{J_r}, \\ b_{32} &= \frac{K_{ls}}{n_g J_g} \end{aligned}$$

The nonlinear character of this system is due to the aerodynamic torque as follows:

$$T_a = \frac{1}{2} \rho \pi R^3 C_q(\lambda, \beta) v^2 \quad (12)$$

This depends on a strongly nonlinear way, the rotor speed  $\omega_r$ , the blade pitch angle  $\beta$  and the wind speed  $v$  which is a not controllable input, random and strongly fluctuating.

### III. DESIGN OF THE FUZZY NEURAL NETWORK SLIDING MODE CONTROL

The main control objective is to maximize the captured wind energy and minimize the efforts of drive train shafts. The power coefficient curve  $C_p(\lambda, \beta)$  has a unique maximum which corresponds to the optimal wind energy [20]:

$$C_p(\lambda_{opt}, \beta_{opt}) = C_{p_{opt}} \quad (13)$$

The rotor, thus, provides maximum aerodynamic power, only to the tip-speed  $\lambda_{opt}$ :

$$\lambda_{opt} = \frac{\omega_{r_{opt}} R}{v} \quad (14)$$

To maximize the captured energy of the wind, the variables  $\lambda$  and  $\beta$  must be maintained at their optimal values in order to ensure maximum value of  $C_p$ . So, the blade pitch angle is fixed at its optimal value  $\beta_{opt}$ . The tip-speed  $\lambda$  depends on both of the wind speed  $v$  and the rotor speed  $\omega_r$ . As the wind speed is not a controllable input, the rotor speed  $\omega_r$  must be adjusted by  $T_{em}$  (As seen in Fig. 2), to track the optimal reference given by:

$$\omega_{r_{opt}} = \frac{\lambda_{opt}}{R} v \quad (15)$$

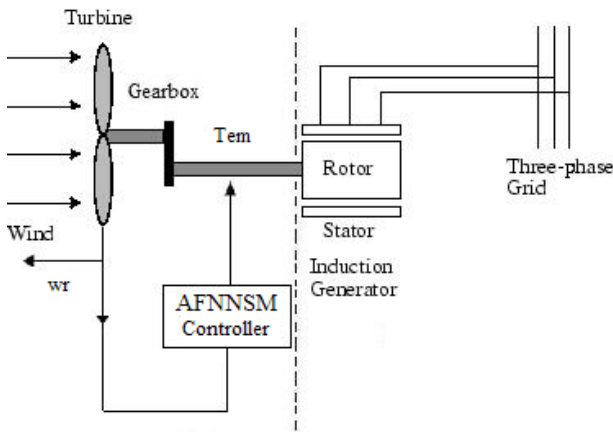


Fig. 2. Basic configuration of wind turbine system.

#### A. Controller Design

Let define  $u = T_{em}$ ,  $x_1 = \omega_r$  and  $x_3 = \omega_g$ , the dynamic system model described by (12) can be rewritten in the state space as follows:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2, x_3) + bu + d(t) \\ \dot{x}_3 = a_{22}x_3 + \frac{a_{23}}{a_{13}}(x_2 - a_{11}x_1 - b_{11}T_a) + b_{32}u \\ y = x_1 \end{cases} \quad (16)$$

$u \in R$  and  $y \in R$  are respectively the input and the output of the system,  $\underline{x} = [x_1, x_2]^T \in R^2$  is the state vector of the system which is assumed to be available for measurement, and  $f(x_1, x_2, x_3)$  is the nominal representation of the system given as:

$$\begin{aligned} f(x_1, x_2, x_3) = & \left( a_{13}a_{31} - \frac{a_{13}a_{33}a_{11}}{a_{13}} \right) x_1 + \left( a_{11} + \frac{a_{13}a_{33}}{a_{13}} \right) x_2 \\ & + a_{13}a_{32}x_3 + \left( a_{13}b_{31} - \frac{a_{13}a_{33}b_{11}}{a_{13}} \right) T_a + b_{11}\dot{T}_a \end{aligned} \quad (17)$$

$d(t)$  represents the unknown model part of the system (uncertainties and external disturbances) and  $b = a_{13}b_{32}$  is known constant.

In the following, the tracking error is defined as:

$$e = \omega_r - \omega_{r_{opt}} \quad (18)$$

The relative degree of the system (16) is  $r = 2$ .

The sliding variable can be defined as:

$$S = \dot{e} + \gamma e = (\dot{x}_2 - \dot{\omega}_{r_{opt}}) + \gamma(x_1 - \omega_{r_{opt}}) \quad (19)$$

Where  $\gamma$  is a positive constant.

Differentiating  $S$  with respect to time, we have:

$$\begin{aligned} \dot{S} &= f(x_1, x_2, x_3) + bu + d(t) + \gamma x_2 - (\ddot{\omega}_{r_{opt}} + \gamma \dot{\omega}_{r_{opt}}) \\ &= F(x, t) + bu \end{aligned} \quad (20)$$

where the function:

$$F(x, t) = f(x_1, x_2, x_3) + \gamma x_2 - (\ddot{\omega}_{r_{opt}} + \gamma \dot{\omega}_{r_{opt}}) + d(t) \quad (21)$$

The following condition (22) must be satisfied [6] to guarantee the existence of sliding mode in finite time:

$$S\dot{S} < -\eta|S| \quad (22)$$

Where  $\eta$  is a small positive constant. The control law that satisfies Eq. (20) is given by:

$$u = u_{eq} + U_s \quad (23)$$

Where  $\hat{u}_{eq} = -\frac{\hat{F}(x, t)}{b}$ , with  $\hat{F}(x, t)$  is the Fuzzy neural network prediction of smooth unknown nonlinear function  $F$ , given in the next section. The additive control is:

$$U_s = -\frac{\alpha}{b} \text{sat } S \quad (24)$$

Where  $\text{sat}(\cdot)$  is the saturation function given by:

$$\text{sat}(S(x, t)) = \begin{cases} S/L & \text{if } |S| < L \\ \text{sgn}(S) & \text{otherwise} \end{cases} \quad (25)$$

$L$  is the boundary layer thickness and  $\alpha$  is chosen according to the following theorem.

The fuzzy neural network prediction error is denoted:

$$F - \hat{F} = \varepsilon_F \quad (26)$$

With  $|\varepsilon_F| < \varepsilon_F^*$  and  $\varepsilon_F^*$  is the upper bound of the network error prediction assumed known.

The block diagram of the AFNNNSM system is depicted in Fig. 3.

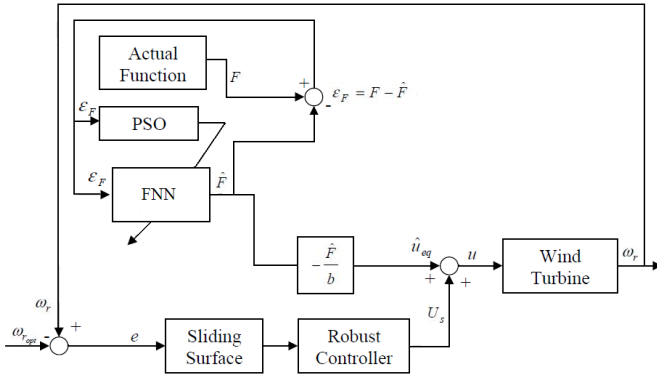


Fig. 3. Block diagram of the proposed AFNNNSM control wind turbine.

**Theorem:** Consider the system described by (16) in the presence of large uncertainties. If the system control is designed as:

$$u = \hat{u}_{eq} - \frac{\alpha}{b} \text{sat}(S) \quad (27)$$

Where  $\hat{u}_{eq} = -\frac{\hat{F}(x, t)}{b}$ , with  $\hat{F}(x, t)$  is the output of the proposed fuzzy neural network,  $\varepsilon_F^* + \eta < \alpha$ ; the trajectory tracking errors will converge to zero in finite time.

**Proof:** Consider the candidate Lyapunov function:

$$V = \frac{1}{2} S^2$$

$$\dot{V} = S\dot{S}$$

$$\dot{V} = S(F(x, t) + bu)$$

By replacing the expression of  $u$  given in the theorem we have:

$$\begin{aligned} \dot{V} &= S(F(x, t) - \hat{F}(x, t) - \alpha \text{sat}(S)) \\ &= S\varepsilon_F - \alpha S \text{sat}(S) \leq |S| |\varepsilon_F| - \alpha S \text{sat}(S) \\ &< |S| \varepsilon_F^* - \alpha S \text{sat}(S) \end{aligned}$$

By choosing  $\varepsilon_F^* + \eta < \alpha$  we have: For any  $L > 0$ , if  $|S| > L$ ,  $\text{sat}(S) = \text{sign}(S)$  the function  $\dot{V} = (\varepsilon_F^* - \alpha)|S| < -\eta|S|$ . However,

in a small L-neighborhood of the origin [6],  $\text{sat}(S) = \frac{S}{L}$  is continuous, the system trajectories are confined to a boundary layer of sliding mode manifold  $S = 0$ .

### B. Fuzzy Neural Network Representation

In this paper, we consider a FNN with five-layered of adjustable weights (Fig. 4). The  $\{x_i\}_{i=1,2}$  are input variables,  $\hat{F} = \hat{\xi}(x, t)$  is the output variable. Fig. 4 shows the structure of the FNN, which is comprised of the input, the membership, the rule, the normalized and the output layers. To give a clear understanding of the signal propagation and mathematical function in each layer, the following section describes FNN functions layer by layer.

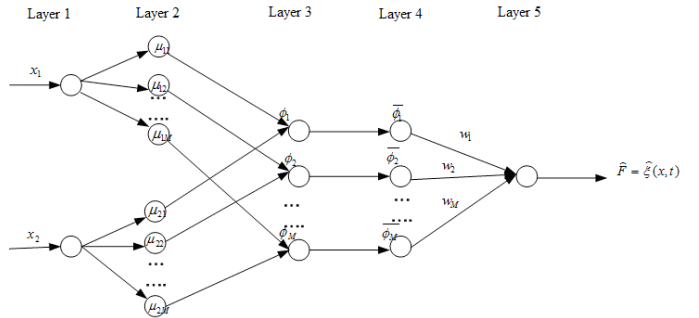


Fig. 4. The architecture of the proposed fuzzy neural network for the prediction of uncertain part.

Layer 1: This layer, which corresponds to one input variable  $x_k$  ( $k = 1, 2$ ), only transmits input values to the next layer directly.

Layer 2: In this layer, each node performs a membership function and acts as a unit of memory. The Gaussian function is adopted as the membership function. For the  $k$ th input, the corresponding net input and output of the  $l$ th node can be expressed as:

$$\mu_k^l(x_k) = \exp\left[-\frac{(x_k - c_k^l)^2}{(\sigma_k^l)^2}\right] \quad (28)$$

where  $c_k^l$  and  $\sigma_k^l$ ;  $l = 1, 2, \dots, M$  are the mean and standard deviation of the Gaussian function of the  $l$ th partition for the  $k$ th input variable  $x_k$ , respectively, and  $M$  is the total number of fuzzy rules.

Layer 3: This layer represents one fuzzy logic rule and performs precondition matching of a rule. The output of a rule node in this layer is calculated by the product operation as follows.

$$\phi_l(x) = \prod_{k=1}^2 \mu_k^l(x_k) \quad (29)$$

Layer 4: In this layer, each of these firing strengths of the rules is compared with the sum of all the firing strengths. Therefore, the normalized firing strengths are computed in this layer as,

$$\bar{\phi}_l(x) = \frac{\phi_l(x)}{\sum_{l=1}^M \phi_l(x)} \quad (30)$$

Layer 5: This layer (or output layer) calculates the summation of its input values from the previous.

$$\hat{\xi}(x) = \sum_{l=1}^M w_l \phi_l(x)$$

The aim of the learning algorithm is to adjust the weights of  $w_\xi^l$ ,  $c_\xi^l$  and  $\sigma_\xi^l$ . The on-line learning algorithm is a gradient descent (GD) search algorithm in the space of network parameters [21]. The essence of (GD) consists of iteratively adjusting the weights the direction opposite to the gradient of error, so as to reduce the discrepancy according to:

$$\dot{w}_l = \eta_{w_l} \varepsilon_F \phi_l \quad (31)$$

Where  $\eta_{w_l}$  is the leaning rate for  $w_l^l$

The updated laws of  $c_\xi^l$  and  $\sigma_\xi^l$  also can be obtained by the gradient decent search algorithm:

$$\dot{c}_k^l = 2\eta_{c_k^l} \varepsilon_F \phi_l w_l \frac{(x_k - c_k^l)}{(\sigma_k^l)^2}$$

$$\dot{\sigma}_k^l = 2\eta_{\sigma_k^l} \varepsilon_F \phi_l w_l \frac{(x_k - c_k^l)^2}{(\sigma_k^l)^3}$$

Where  $\eta_{c_k^l}$  and  $\eta_{\sigma_k^l}$  are the learning-rate parameters of the mean and the standard deviation of the Gaussian functions.

### C. Particle Swarm Optimization Algorithm

The objective is to optimize the learning rate  $\eta_{w_l}$  by the PSO algorithm.

In PSO,  $m$  particles fly through an  $n$ -dimensional search space. For each particle  $i$ , there are two vectors: the velocity vector  $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$  and the position vector  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ . Similar to bird socking and fishes schooling, the particles are updated according to their previous best position  $P_i = (p_{i1}, p_{i2}, \dots, p_{in})$  and the whole swarm's previous best position  $P_g = (p_{g1}, p_{g2}, \dots, p_{gn})$ . This means that particle  $i$  adjusts its velocity  $V_i$  and position  $X_i$  in each generation according to the equations bellow [22]:

$$v_{id}(t+1) = v_{id}(t) + c_1 \times rand()_1 \times (p_{id} - x_{id}) + c_2 \times rand()_2 \times (p_{gd} - x_{id}) \quad (32)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (33)$$

Where  $d = 1, 2, \dots, n$ ;  $c_1, c_2$  are the acceleration coefficients with positive values, which we take as:  $c_1 = 1$  and  $c_2 = 2$  in this paper;  $rand()_1, rand()_2$  are random numbers between 0 and 1. The new velocity and position for each particle are calculated using the equations (32) and (33) based on its velocity  $v_{id}(t)$ , best position  $P_{id}$  and the swarm's best position  $P_{gd}$ .

In order to calculate the optimized parameter of learning rate  $\eta_{w_l}$  given in equation (29), the PSO is used *on-line* to minimize the neural network prediction error.

We define the quadratic errors  $e_{rq}$  as:  $e_{rq}(t_i) = \varepsilon_F^2(t_i)$

The objective function  $f$  to be minimized is chosen as the norm of the quadratic error:  $f = norm(E_{rq})$  with  $E_{rq}$  the vector that contains all errors  $e_{rq}(t_i)$ .

The PSO algorithm tests the search space using  $m$  particles

according to (30) and (31). Each particle  $i$  moves in search space and stores its best position  $P_{id}$ , then, it compares all positions to finally take out the chosen  $\eta_{k-optimum}$  that gives the minimum value of the objective function  $f$ .

## IV. SIMULATION RESULTS

Different simulation cases are considered for the case of a two-mass model of the wind turbine with:

- Presence of constant additive control disturbance  $d$  of 10  $kN.m$ ,
- Presence of an additive measurement noise on  $\omega_g$ , with a SNR of approximatively 7  $dB$ .
- A wind speed profile of 7 m/s means value.

The two-mass model parameters are presented below:

$$R = 21.65 \text{ m}, \rho = 1.29 \text{ kg/m}^3,$$

$$J_r = 3.25 \times 10^5 \text{ kg.m}^2, J_g = 34.4 \text{ kg.m}^2,$$

$$K_{ls} = 9500 \text{ N.m/rad/s}, B_{ls} = 2.691 \times 10^5 \text{ N.m/rad},$$

$$K_r = 27.36 \text{ N.m/rad/s}, K_g = 0.2 \text{ N.m/rad/s},$$

$$n_g = 43.165.$$

Three membership functions have been used for each of the two inputs of the FNN. The associated fuzzy sets with Gaussian membership function for each input signal, have initial input for the mean and the standard deviation:

$$c_1 = [-6.4470 \quad 1.2640 \quad 21.9693]$$

$$\sigma_1 = 1.0e + 005 * [0.0043 \quad 0.0241 \quad 2.1675]$$

$$c_2 = [-130.1245 \quad -275.7842 \quad -30.8045]$$

$$\sigma_2 = 1.0e + 005 * [0.0140 \quad 0.0261 \quad 4.1794]$$

The initial output weights :

$w = [-10.9070 \quad 119.7829 \quad -118.3859]$ , being the initial weights, which are obtained hors-line identification system.

Fig. 5 presents the wind speed profile used in this study. Fig. 6 shows the estimation error for the predicting function F.

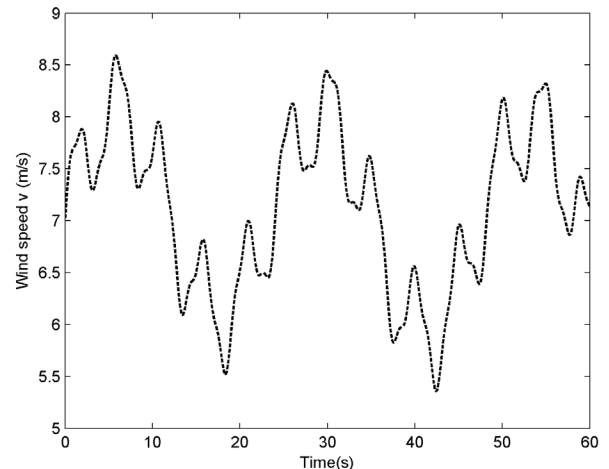


Fig. 5. Wind speed profile of 7 m/s mean value.

Fig. 5 is proposed for an average speed, it was considered the case where the wind speed is highly variable and the case where the wind speed does not change rapidly.

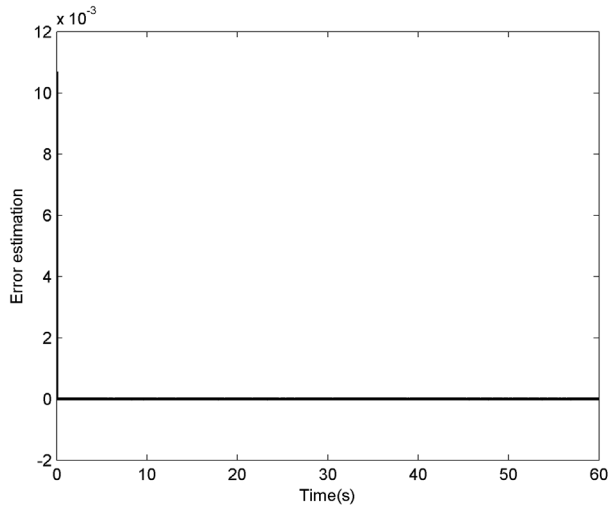


Fig. 6. Error estimation of the predicting function F.

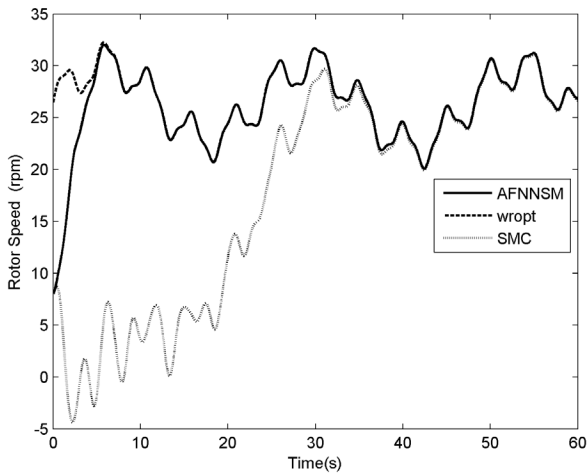


Fig. 7. Rotor speed with  $\alpha = 1.2$ .

To gain a large enough  $\alpha = 1.2$  we see the output controlled by the standard sliding mode converges slowly. However that commissioned by the proposed method NNSMC converges rapidly to the desired output

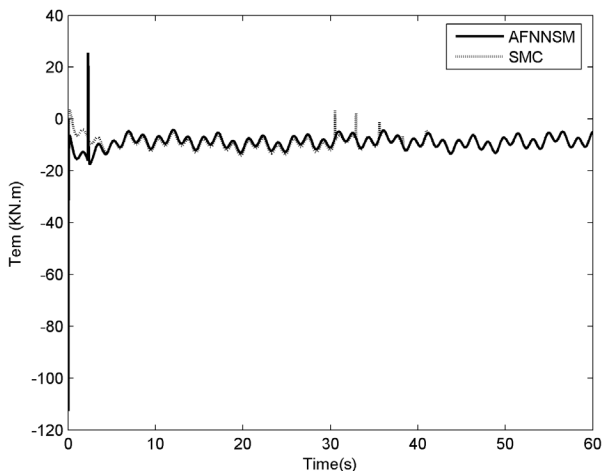


Fig. 8. Electromagnetic torque with  $\alpha = 1.2$ .

Fig. 8 shows the actual and optimal trajectory of the rotor speed, we find that the best tracking performance is obtained when applying the RFNNNSM.

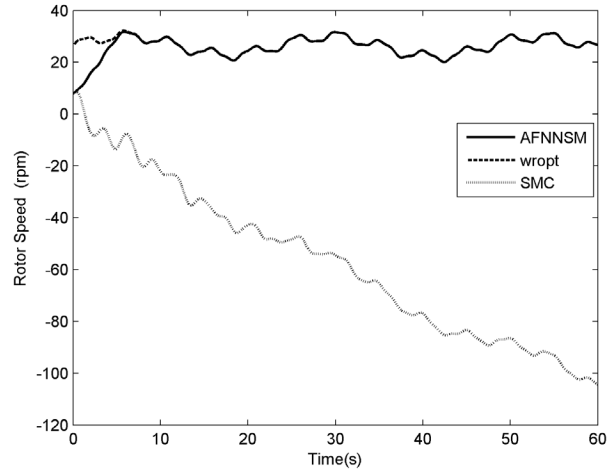


Fig. 9. Rotor speed with  $\alpha = 0.9$ .

For comparison we have considered in the control law, for both AFNNNSM and traditional SMC controllers, the same gain  $\alpha = 1.2$  (see the obtained outputs in Fig. 7) and  $\alpha = 0.9$  (see the obtained outputs in Fig. 9). From these figures, it can be seen that, the best tracking performance is obtained when the proposed AFNNNSM controller is applied. The corresponding control signals are given in Fig. 8 for  $\alpha = 1.2$ , and in Fig. 10 for  $\alpha = 0.8$ . Fig. 9 shows that the standard SMC controller totally failed to control the system.

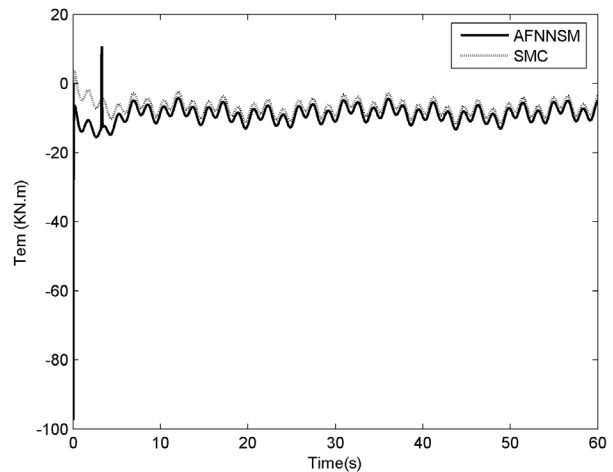


Fig. 10. Electromagnetic torque with  $\alpha = 0.9$ .

By using the a gain  $\alpha = 0.9$ , Fig. 10 shows the optimal trajectory of the rotor speed, otherwise we find that the best tracking performance is obtained when the RFNNNSM is applying.

Table 1 contains both the gain used in this study as well as the number of the iteration in order to find the global optimum particle of learning rate, so we can remark that the global optimum is found from the 21<sup>st</sup> iteration.



TABLE I. THE GLOBAL OPTIMUM PARTICLE OF LEARNING RATE

Iteration number	$\eta_{w_i}$ for $\alpha = 0.9$	$\eta_{w_i}$ for $\alpha = 1.2$
1	0.7395	0.4492
2	0.7100	0.4665
3	0.6821	0.4804
4	0.6745	0.4940
5	0.6745	0.5156
6	0.6794	0.5264
7	0.6897	0.5230
8	0.6997	0.5142
9	0.7035	0.5092
10	0.7011	0.5102
11	0.6967	0.5113
12	0.6953	0.5119
13	0.6949	0.5121
14	0.6954	0.5139
15	0.6956	0.5145
16	0.6968	0.5147
17	0.6975	0.5143
18	0.6976	0.5138
19	0.6975	0.5136
20	0.6972	0.5137
21	0.6971	0.5138
22	0.6971	0.5138
23	0.6972	0.5138
24	0.6972	0.5138
25	0.6972	0.5138
26	0.6972	0.5138
27	0.6972	0.5138
28	0.6972	0.5138
29	0.6972	0.5138
30	0.6972	0.5138

## V. CONCLUSION

This paper addressed the robust optimal reference tracking problem for a variable speed wind turbine. The designed method is a combination of traditional sliding mode control approach and fuzzy neural network. The later is employed to approximate the unknown nonlinear model function with online adaptation of parameters via BP learning algorithm. This provides a better description of the plant, and hence enables a lower switching gain to be used despite the presence of large uncertainties. The particle swarm optimization (PSO) algorithm is used to optimize the learning rate of BP algorithm in order to get faster convergence. The comparison with the traditional sliding mode control has been realized and simulation results have shown a good performance of the proposed method to track the optimal reference without any chattering problem.

## REFERENCES

- [1] J. F. Manwell, J. G. McGowan, A. L. Rogers, Wind Energy Explained: Theory, Design and Applications (John Wiley & Sons, 2002).
- [2] G. Ofualagba, E. U. Ubeku, Wind energy conversion system- wind turbine modelling, Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, IEEE, 2008, 1-8.
- [3] T. Burton, D. Sharpe, N. Jenkins, E. Bossanyi, Wind Energy Handbook (John Wiley & Sons, 2001).
- [4] S. Abdeddaim, A. Betka. Optimal Tracking and Robust Power Control of the Doubly Fed Induction Generator wind Turbine, Electric Power Systems Research, vol. 49, 2013, pp. 234–242.
- [5] V. I. Utkin, Sliding Modes in Control Optimization (Springer-Verlag, 1992).
- [6] J.J. Slotine, Sliding Controller Design for Non-linear Systems, International Journal of Control, vol. 40, n. 2, 1984, pp. 421–434.
- [7] V. Utkin, J. Guldner, J. Shi, Sliding Mode Control in Electromechanical System (Taylor & Francis, London, 1999).
- [8] P.F. Puleston, F. Valenciaga, Chattering Reduction in a Geometric Sliding Mode method. A Robust Low-chattering Controller for an Autonomous Wind System, Control and Intelligent Systems, vol. 37, n. 1, 2009, pp. 39-45.
- [9] H. Amimeur, D. Aouzellag, R. Abdessemed, K. Ghedamsi, Sliding Mode Control of a Dual-stator Induction Generator for Wind Energy Conversion Systems, International Journal of Electrical Power and Energy Systems, vol. 42, n. 1, 2012, pp. 60–70.
- [10] Y. C. Chen, C. C. Teng, A Model Reference Control Structure Using a Fuzzy Neural Network. Fuzzy Sets and Systems, vol. 73, n. 3, 1995, pp. 291–312.
- [11] V. Gorrini, H. Bersini, Recurrent fuzzy systems. In Proceedings of the IEEE international conference on fuzzy systems, vol. 1, 1994, 193–198.
- [12] F.B. Duh, C.T. Lin, Tracking a Maneuvering Target Using Neural Fuzzy Network, IEEE Transactions on Systems, Man, and Cybernetics, vol. 34, n. 1, 2004, pp. 16–33.
- [13] C.F. Juang, C.H. Hsu, Temperature Control by Chip-implemented Adaptive Recurrent Fuzzy Controller Designed by Evolutionary Algorithm, IEEE Transactions on Circuits System, vol. 52, n. 11, 2005, pp. 2376–2384.
- [14] S.J. Lee, C.L. Hou, A Neural-fuzzy System for Congestion Control in ATM Networks, IEEE Transactions on Systems, Man, and Cybernetics, vol. 30, n. 1, 2000, pp. 2–9.
- [15] J.M. Rubio, & LT. Aguilar, Maximizing the Performance of Variable Speed Wind Turbine with Nonlinear Output Feedback Control, Procedia Engineering, vol. 35, 2012, pp. 31-40.
- [16] L. Munteanu, A. L. Bratcu, N-A. Cutululis, E. Ceanga, Optimal Control of Wind Energy Systems, Advances in Industrial Control (Springer, 2008).
- [17] J.F. Adria, G.B. Oriol, S. Andreas, S. Marc, M. Montserrat, Modeling and Control of The Doubly Fed Induction Generator Wind Turbine, Simulation Modelling Practice and Theory, vol. 18, n. 9, 2010, pp. 1365–1381.
- [18] Ph. Delarue, A. Bouscayrol, A. Tounzi, X. Guillaud, Modelling, Control and Simulation of an Overall Wind Energy Conversion System, Renewable Energy, vol. 28, n. 8, 2003, pp. 1169-1185.
- [19] A. Gaillard, P. Poure, S. Saadate, M. Machmoum, Variable Speed DFIG Wind Energy System for Power Generation and Harmonic Current Mitigation, Renewable Energy, vol. 34, n.6, 2009, pp. 1545–1553.
- [20] K. A. Stol, Geometry and Structural Properties for the Controls Advanced Research Turbine (CART) from Model Tuning (Subcontractor Report SR-500-32087, National Renewable Energy Laboratory, Golden, CO, September, 2004).
- [21] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland (Ed), Parallel Distributed Processing, (Cambridge, MA, (1): MIT Press, 1986, 318–362).
- [22] M. A. Cavuslu, C. Karakuzub, Neural Identification of Dynamic Systems on FPGA with Improved PSO Learning. Applied Soft Computing, vol. 12, 2012, pp. 2707–2718.
- [23] Esmailpour, M., and A. R. A. Mohammadi, Analyzing the EEG Signals in Order to Estimate the Depth of Anesthesia using Wavelet and Fuzzy Neural Networks, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 4, n.2, 2016, pp. 12-15. Doi: 10.9781/ijimai.2016.422.



Nabil Farhane

Nabil Farhane received his M.Sc. degree from Sidi Mohammed ben Abdellah University of Fez in 2011. He is currently pursuing the Ph.D. degree at Sidi Mohammed ben Abdellah University. His research interests include fuzzy logic, neural networks, non-linear control, etc.



Ismail Boumhidi

Ismail Boumhidi is a professor of electronics at the Faculty of Sciences, Fez, Morocco. He received his Ph.D. degree from Sidi Mohammed ben Abdellah University, Faculty of Sciences, in 1999. His research areas include adaptive robust control, multivariable non-linear systems and fuzzy logic control with applications.



Jaouad Boumhidi

Jaouad Boumhidi received his Ph.D. degree in Automatic Control and Computer Sciences from Sidi Mohammed ben Abdellah University in 2005. He is now a professor at Sidi Mohammed ben Abdellah University. His research interests include neural networks, fuzzy logic, robust control, etc.