

Research Article

Application of a Data-Driven Fuzzy Control Design to a Wind Turbine Benchmark Model

Silvio Simani

Department of Engineering, University of Ferrara, 44122 Ferrara, Italy

Correspondence should be addressed to Silvio Simani, silvio.simani@unife.it

Received 1 August 2012; Accepted 2 November 2012

Academic Editor: Sendren Sheng-Dong Xu

Copyright © 2012 Silvio Simani. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In general, the modelling of wind turbines is a challenging task, since they are complex dynamic systems, whose aerodynamics are nonlinear and unsteady. Accurate models should contain many degrees of freedom, and their control algorithm design must account for these complexities. However, these algorithms must capture the most important turbine dynamics without being too complex and unwieldy, mainly when they have to be implemented in real-time applications. The first contribution of this work consists of providing an application example of the design and testing through simulations, of a data-driven fuzzy wind turbine control. In particular, the strategy is based on fuzzy modelling and identification approaches to model-based control design. Fuzzy modelling and identification can represent an alternative for developing experimental models of complex systems, directly derived directly from measured input-output data without detailed system assumptions. Regarding the controller design, this paper suggests again a fuzzy control approach for the adjustment of both the wind turbine blade pitch angle and the generator torque. The effectiveness of the proposed strategies is assessed on the data sequences acquired from the considered wind turbine benchmark. Several experiments provide the evidence of the advantages of the proposed regulator with respect to different control methods.

1. Introduction

Wind turbines are complex nonlinear dynamic systems forced by gravity and stochastic wind disturbance, which are affected by gravitational, centrifugal, and gyroscopic loads. Their aerodynamics are nonlinear and unsteady, whilst their rotors are subject to complicated turbulent wind inflow fields driving fatigue loading. Therefore, wind turbine modelling and control represent complex and challenging tasks [1, 2].

Accurate models have to contain many degrees of freedom in order to capture the most important dynamic effects. It is clear that the design of control algorithms for wind turbines has to take into account these complexities. On the other hand, control algorithms must capture the most important turbine dynamics, without being too complex and unwieldy [1, 3].

Today's wind turbines employ different control actuation and strategies to achieve the required goals and performances. Some turbines perform the control action through passive methods, such as in fixed-pitch, stall control

machines. In these machines, the blades are designed so that the power is limited above rated wind speed through the blade stall. Thus, no pitch mechanism is needed [1]. On the other hand, below rated wind speed, the generator speed is fixed [4]. Rotors with adjustable pitch are often used in constant-speed machines to provide turbine power control better than the one achievable with blade stall [5]. Therefore, blade pitching can be regulated to provide constant power above rated wind speed. The pitch mechanisms have to be quite fast, in order to provide good power regulation in the presence of gusts and turbulence. In order to maximise the power output below the wind speed, the rotational speed of the turbine must vary with wind speed.

The control strategies for such machines are typically designed using simple classical control design techniques such as (Proportional Integral Derivative) PID control for blade pitch regulation, as shown for example, in [6]. Advanced controls can be used to improve the energy capture, for example through the generator torque control, as shown for example, in [7]. In the same situation,

Johnson et al. [8] proposed an adaptive control approach. Advanced multivariable control design methods, such as those based on state-space models, can be used to meet control objectives and use all the available actuators and sensors in a reduced number of control loops. In [9], a multivariable approach was used to design both an independent pitch controller to mitigate the effects of asymmetric wind disturbances. Even though two separate control loops were used for example, in [9], the multivariable control design approach can require fewer control loops compared to classical control design methods, as shown for example, in [10]. Therefore, the trade-off between wind turbine control algorithms' accuracy and their limited computational complexity represents the challenging point that motivates this study.

The first contribution of this work consists of providing an application example of the design, and testing through simulations, of a data-driven fuzzy wind turbine control. In particular, the strategy is based on fuzzy modelling and identification approaches to model-based control design. As stated above, since a mathematical model is a description of system behaviour, accurate modelling for a complex nonlinear system can be very difficult to achieve in practice. Thus, fuzzy modelling and identification can represent an alternative for developing experimental models of complex systems, such as wind turbine systems considered in this work. In contrast to pure nonlinear identification methods, fuzzy systems are capable of deriving nonlinear models directly from measured input-output data without detailed system assumptions [11]. Thus, it is suggested to describe the plant under investigation by a collection of local affine systems of the type of Takagi-Sugeno (TS) fuzzy prototypes [12], whose parameters are obtained by identification procedures. The proposed identification approach is motivated by previous works by the same author [13, 14].

Regarding the controller design, this paper suggests a fuzzy control approach for the adjustment of both the wind turbine blade pitch angle and the generator torque, with application to a wind turbine benchmark. This design is performed according to the following steps. Firstly, a PI regulator is devised using the classic Ziegler-Nichols method [15]. Then, the corresponding fuzzy (Proportional Integral) PI controller is built, by means of a suitable choice of the gains. The membership functions (MFs) and rules are derived directly from the identified TS fuzzy models. The effectiveness of the proposed fuzzy modelling and control strategy are assessed on the data sequences acquired from the considered benchmark. Several simulation experiments provide the evidence of the advantages of the proposed fuzzy PI regulator with respect to the switching control strategy developed in [16].

The capabilities of the proposed control strategy are compared with respect to the control strategy developed in [16], and with reference to different control methods based for example, on sliding mode or neural controllers [17, 18].

The next feature of the work is related with the impact evaluation of the modelling uncertainties, disturbance, and measurement errors on the designed control scheme. In particular, the paper proposes the use of extensive Monte-Carlo simulations for the analysis and the assessment of the

design, the robustness, the stability, and its final performance evaluation. In fact, as in this case the wind turbine cannot be described by any analytical dynamic model, the Monte-Carlo tool represents the only method for evaluating the performances of the developed control scheme when applied to the monitored process.

The last key point of the paper concerns the assessment of the developed control algorithms through a hardware in the loop (HIL) facility, which has been developed to qualify the proposed approaches in the presence of experimental limitations closer to real situations.

Finally, the paper has the following structure. Section 2 provides an overview of the wind turbine system considered in this work. Section 3 recalls the fuzzy modelling and identification strategy exploited for the design of the fuzzy controller. This proposed fuzzy controller design, followed by a simple tuning strategy, are presented in Section 3.2. The achieved results and comparisons with different control strategies are summarised in Section 4, where the stability, the robustness analysis, and the capabilities of the developed control method with respect to measurement and modelling errors are investigated in simulation. Realistic simulations and comparisons with different control schemes relying on sliding mode and neural controller are also reported. Section 5 ends the paper by highlighting the main achievements of the work, providing some suggestions for possible further research topics.

2. Wind Turbine Benchmark Description

The three-blade horizontal axis turbine considered in this paper works according to the principle that the wind is acting on the blades, and thereby moving the rotor shaft. In order to up-scale the rotational speed to the needed one at the generator, a gear box is introduced. A more accurate description of the benchmark model can be found in [16, 19]. The diagram of the wind turbine model is sketched in Figure 1.

The rotational speed, and consequently the generated power, is regulated by means of two controlled inputs: the converter torque $\tau_g(t)$ and the pitch angle $\beta_r(t)$ of the turbine blades. From the wind turbine system, a number of measurements can be acquired. $\omega_r(t)$ is the rotor speed, $\omega_g(t)$ is the generator speed, and $\tau_g(t)$ is the torque of the generator controlled by the converter, which is provided with the torque reference, $\tau_r(t)$. The estimated aerodynamic torque is defined as $\tau_{aero}(t)$. This estimate clearly depends on the wind speed, which unfortunately is very difficult to measure correctly. A very uncertain measurement can be available as described in [16].

2.1. Model Description. This section recalls briefly the wind turbine model description in the continuous-time domain. It is subsequently approximated via identified discrete-time prototypes, as shown in Section 3.

The aerodynamic model is defined as in

$$\tau_{aero}(t) = \frac{\rho A C_p(\beta_r(t), \lambda(t)) v^3(t)}{2\omega_r(t)}, \quad (1)$$

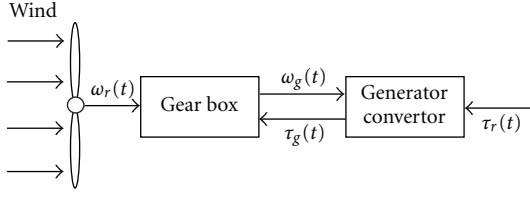


FIGURE 1: Logic diagram of the wind turbine, where only two of the three blades are shown.

where ρ is the density of the air, A is the area covered by the turbine blades in its rotation, $\beta_r(t)$ is the pitch angle of the blades, $v(t)$ is the wind speed, whilst $\lambda(t)$ is the tip-speed ratio of the blade, defined as

$$\lambda(t) = \frac{\omega_r(t)R}{v(t)} \quad (2)$$

with R the rotor radius. C_p represents the power coefficient, here described by means of a two-dimensional map (look up table) [16]. Equation (1) is used to estimate $\tau_{aero}(t)$ based on an assumed estimated $v(t)$, and measured $\beta_r(t)$ and $\omega_r(t)$. Due to the uncertainty of the wind speed, the estimate of $\tau_{aero}(t)$ is considered affected by an unknown measurement error, which motivates the approaches described in Section 3.

A simple one-body model is used to represent the drive train, as described in [19]. The generator torque $\tau_g(t)$ and the reference $\tau_r(t)$ are in this context transformed to the low speed side of the drive train (rotor side), whilst p_{gen} is the generator power coefficient. With these assumptions, the complete continuous-time description of the system under diagnosis has the form of

$$\begin{aligned} \dot{x}_c(t) &= f_c(x_c(t), u(t)), \\ y(t) &= x_c(t), \end{aligned} \quad (3)$$

where $u(t) = [\beta_r(t), \tau_g(t)]^T$ and $y(t) = x_c(t) = [P_g(t), \omega_g(t)]^T$ are the control inputs and the monitored output measurements, respectively. $f_c(\cdot)$ represents the continuous-time nonlinear function that will be approximated via discrete-time prototypes from N sampled data $u(k)$ and $y(k)$, with $k = 1, 2, \dots, N$, presented in Section 3.

Regarding the input and output signals, $\omega_g(t)$ is the generator speed measurement, $P_g(t)$ is the generator power measurement, $\beta_r(t)$ is the pitch measurement, and $\tau_g(t)$ is generator torque measurement. Finally, the model parameters and the map $C_p(\beta, \lambda)$ are chosen in order to represent a realistic turbine, which is used as benchmark system in this study [16].

2.2. Wind Turbine Benchmark Control System. The controller for a wind turbine operates in principle in two main zones. Zone 1 is the power optimisation, whilst zone 2 corresponds to constant power production. The optimal power in zone 1 is obtained if the blade pitch angle β_r is equal 0 degrees, and if the tip speed ratio is constant at its optimal value λ_{opt} . The tip speed ratio, λ , as already described by (2), can be written

as in (4), where R is the radius of the blades, v is the wind speed, and ω_r is the angular rotor speed:

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

The optimal value of λ , which is denoted with λ_{opt} , is determined as the maximal value point in the power coefficient mapping of the wind turbine. This optimal power is achieved by setting the reference torque to the converter, $\tau_g = \tau_{gr}$. The reference torque τ_{gr} in this power optimisation zone can be written as

$$\tau_{gr} = K_{opt} \omega_r^2 \quad (5)$$

with:

$$K_{opt} = \frac{1}{2} \rho A R^3 \frac{C_{p_{max}}}{\lambda_{opt}^3} \quad (6)$$

with ρ being the air density, A the area swept by the turbine blades, $C_{p_{max}}$ the maximal value of C_p (i.e., the power coefficient map), related to the λ_{opt} , that is, the optimal tip-speed ratio.

When the power reference is achieved, the controller can be switched to control zone 2. In this zone the control objective consists of following the power reference, P_r , which is obtained by controlling β_r , such that the C_p is decreased. In a traditional industrial control scheme, usually a PI controller is used to keep ω_r at the prescribed value by changing β_r . The second control input is τ_g , whose value is computed by using (5).

The wind turbine benchmark controller considered in this study was implemented with a sample frequency at 100 Hz, that is, $T_s = 0.01$ s. The controller starts in zone 1. Therefore, the control mode should switch from zone 1 to zone 2 if the following conditions hold [16]:

$$P_g(k) \geq P_r(k), \quad \omega_g(k) \geq \omega_{nom}, \quad (7)$$

where k indicates the acquired discrete-time measurements from the corresponding continuous-time signals, whilst ω_{nom} is the nominal turbine speed. On the other hand, the control switches from zone 2 to zone 1 if

$$\omega_g(k) < \omega_{nom} - \omega_{\Delta}, \quad (8)$$

where ω_{Δ} is a number that introduces hysteresis to ensure a minimum time between transitions. On the other hand, for the control zone 2:

$$\beta_r(k) = \beta_r(k-1) + k_p e(k) + (k_i T_s - k_p) e(k-1), \quad (9)$$

$$e(k) = \omega_g(k) - \omega_{nom}$$

with $k = 1, 2, \dots, N$. The parameters for this PI *speed controller* are $k_i = 0.5$ and $k_p = 3$, with sampling time $T_s = 0.01$ s., as described in [19].

Finally, regarding the control of the input τ_g , a second PI regulator is implemented, as the one of (9):

$$\tau_r(k) = \tau_r(k-1) + k_p e(k) + (k_i T_s - k_p) e(k-1), \quad (10)$$

$$e(k) = P_g(k) - P_r.$$

The parameters for this second PI *power controller* are $k_i = 0.014$ and $k_p = 447 \times 10^{-6}$, according to [19].

3. Fuzzy Identification and Control Design

This section describes the comprehensive approach exploited for obtaining the fuzzy description of the wind turbine system and the control strategy used for the regulation of its input variables. In particular, the fuzzy modelling and identification scheme, which is recalled in Section 3.1, enhances the design procedure of the proposed fuzzy controller, as shown in Section 3.2.

3.1. Fuzzy Identification and Data Clustering. In order to generate fuzzy models automatically from measurements, a comprehensive methodology is used. This employs fuzzy clustering techniques to partition the available data into subsets characterised by a linear behaviour. The relationships between the presented identification method and linear regression are exploited, allowing for the combination of fuzzy logic techniques with system identification tools. In addition, the implementation in the Matlab toolbox of the Fuzzy Modelling and Identification (FMID) techniques presented in the following is available [20]. Fuzzy identification usually refers to methods and algorithms for constructing fuzzy models from data.

A large part of fuzzy modelling and identification algorithms (see, e.g., [11] and references therein) share a common two-step procedure, in which at first, the operating regions are determined using heuristics or data clustering techniques. Then, in the second stage, the identification of the parameters of each submodel is achieved using the identification algorithm in particular proposed by the author [13, 14], which can be seen as a generalisation of classical least-squares. From this perspective, fuzzy identification can be regarded as a search for a decomposition of a nonlinear system, which gives a desired balance between the complexity and the accuracy of the model, effectively exploring the fact that the complexity of systems is usually not uniform. A suitable class of fuzzy clustering algorithms can be thus used for this decomposition purpose, and in particular, the well-known Gustafson-Kessel (GK) fuzzy clustering [11] is exploited in this work, since already implemented and available in [20].

The fuzzy rule-based model suitable for the approximation of a large class of nonlinear systems was introduced by Takagi and Sugeno (TS) in [12]. In the TS fuzzy model, the rule consequents are crisp functions of the model inputs:

$$R_i : \mathbf{IF} \mathbf{x}(k) \text{ is } A_i \text{ THEN } y_i = f_i(\mathbf{x}(k)), \quad (11)$$

where $i = 1, 2, \dots, M$, $\mathbf{x}(k) \in \mathfrak{R}^p$ is the input (antecedent) variable and $y_i \in \mathfrak{R}$ is the output (consequent) variable. R_i denotes the i th rule, and M is the number of rules in the rule base. A_i is the antecedent fuzzy set of the i th rule, defined by a (multivariate) membership function. The consequent functions f_i are typically chosen as instances of a suitable parameterised function, whose structure remains equal in all the rules and only the parameters vary. A simple and practically useful parameterisation of the function f_i is the affine form:

$$y_i = \mathbf{a}_i \mathbf{x} + b_i, \quad (12)$$

where \mathbf{a}_i is the parameter vector (regressand), and b_i is the scalar offset. $\mathbf{x} = \mathbf{x}(k)$ represents the regressor vector, which can contain delayed samples of $u(k)$ and $y(k)$. This model is referred to as *affine TS model*, and it can be written as [12]

$$y = \frac{\sum_{i=1}^M \mu_i(\mathbf{x}) y_i}{\sum_{i=1}^M \mu_i(\mathbf{x})}. \quad (13)$$

The *antecedent* fuzzy sets μ_i are extracted from the fuzzy partition matrix [11, 20]. The *consequent parameters* \mathbf{a}_i and b_i are estimated from the data using the method developed by the author [13, 14] and recalled below. This identification scheme exploited for the estimation of the TS model parameters has been integrated into the FMID toolbox for Matlab by the author. This approach developed by the author is usually preferred when the TS model should serve as predictor, as it computes the consequent parameters by the so-called *Frisch scheme* [13, 14]. Therefore, after the clustering of the data has been obtained via the GK algorithm, the data subsets are processed according the Frisch scheme identification procedure [13, 14], in order to estimate the TS parameters for each affine submodels.

3.2. Fuzzy Controller Design. The proposed fuzzy logic controller is fed by the error signal $e(k)$, that is, the tracking error defined as the difference between the considered set-point $r(k)$ and the plant controlled output $y(k)$ at the sample k :

$$e(k) = r(k) - y(k). \quad (14)$$

The fuzzy PI controller uses a second input signal, defined as the sum of the system errors, which is computed using

$$\delta e(k) = \sum_{i=1}^k e(i). \quad (15)$$

It is known from the Digital Control Theory that the most frequently used digital PI control algorithm can be described as follows:

$$u(k) = k_p e(k) + k_i \delta e(k), \quad (16)$$

where $k_i = k_p(T_s/T_i)$, T_s is the sampling time, T_i is the integral time constant of the conventional controller, k_p is the proportional gain, and $u(k)$ is the output control action.

The structure of the control system with the proposed fuzzy controllers are based on Sugeno's fuzzy technique. The Sugeno's fuzzy rules into the fuzzy PI controller are in the generalised form of "IF-THEN" composition with a premise and an antecedent part to describe the control policy. The rule base comprises a collection of M rules, where the index j represents the rule number:

$$R_j : \mathbf{IF} \mathbf{x}(k) \text{ is } A_j \text{ THEN}$$

$$f_u^{(j)}(k) = K_p^{(j)} e(k) + K_i^{(j)} \delta e(k) + b_j, \quad j = 1, 2, \dots, M, \quad (17)$$

where $e(k)$ and $\delta e(k)$ are the input variables, with $k = 1, \dots, N$. In this expression, a similarity between the expression of the conventional digital PI controller of (16) and the Sugeno's output function of (17) can be found. In this case, the fuzzy PI controller is considered as a collection of several local PI controllers, which are represented by the Sugeno's functions into the different fuzzy rules.

For a discrete universe with M quantisation levels in the fuzzy output, the control action $u = u_F$ is expressed as a weighted average of the Sugeno's output functions f_u , and their membership degrees μ_i of the quantisation levels, with $i = 1, \dots, M$. Also in this case, before the output can be inferred, the degree of fulfilment of the antecedent denoted by $\mu_i(\mathbf{x})$ must be computed. Thus, the degree of fulfilment is simply equal to the membership degree of the given input \mathbf{x} , that is, $\mu_i = \mu_{A_i}(\mathbf{x})$. By recalling the identified TS model, the inference is reduced to a simple expression, similar to the fuzzy-mean defuzzification formula [11]:

$$u_F = \frac{\sum_{j=1}^M \mu_j(\mathbf{x}) f_u^{(j)}}{\sum_{j=1}^M \mu_j(\mathbf{x})} \quad (18)$$

or by substituting the expression of the fuzzy PI terms:

$$u_F(k) = \frac{\sum_{j=1}^M \mu_j(\mathbf{x}(k)) (K_P^{(j)} e(k) + K_I^{(j)} \delta e(k) + b_j)}{\sum_{j=1}^M \mu_j(\mathbf{x}(k))}, \quad (19)$$

where the time dependence at the instant k has been highlighted.

It is worth noting that the PI controller parameters $K_P^{(j)}$ and $K_I^{(j)}$ (with $j = 1, \dots, M$) are settled according to the Ziegler-Nichols rules applied to the identified local linear TS submodels. Then, in order to obtain a quick reaction to set-point variations, gain scheduling of the fuzzy regulator parameters is performed depending on the error, as shown by (19).

In more detail, if for example the TS consequents are represented by second order discrete-time local linear dynamic models ($n = 2$) described in (12) by their identified parameters $\mathbf{a}_i = [\alpha_1^{(i)}, \alpha_2^{(i)}, \gamma_1^{(i)}, \gamma_2^{(i)}]$, and regressor $\mathbf{x}^T(k) = [-\gamma(k-1), -\gamma(k-2), u(k-1), u(k-2)]$, ($i = 1, \dots, M$), the so-called critical gain $K_o^{(i)}$ and critical period of oscillations $T_o^{(i)}$ required by the Ziegler-Nichols method can be computed for example, as follows [15]:

$$K_o^{(i)} = \frac{\alpha_1^{(i)} - \alpha_2^{(i)} - 1}{\gamma_2^{(i)} - \gamma_1^{(i)}}, \quad (20)$$

$$T_o^{(i)} = \frac{2\pi T_s}{\arccos \delta^{(i)}}, \quad \text{with } \delta^{(i)} = \frac{\alpha_2^{(i)} \gamma_1^{(i)} - \alpha_1^{(i)} \gamma_2^{(i)}}{2\gamma_2^{(i)}}$$

which holds if $|\delta^{(i)}| < 1$. Alternative formulas can also be used [15]. Equation (21) is thus used for calculating the

parameters $K_P^{(i)}$ and $K_I^{(i)}$ of the (local) i th PI controller of (17):

$$K_P^{(i)} = 0.6 K_o^{(i)} \left(1 - \frac{T_s}{T_o^{(i)}} \right), \quad (21)$$

$$K_I^{(i)} = \frac{1.2 K_o^{(i)}}{K_P^{(i)} T_o^{(i)}},$$

where T_s is the sampling time. As shown for example, in [15], TS consequents of orders n greater than 2 can be considered.

Once the fuzzy PI controller parameters have been computed, the second step consists of building the fuzzy controller of (19). The input MFs $\mu_j(\mathbf{x})$ can coincide with the ones of the identified TS model, as described for example, in [11]. The number of the input MFs determines the number of rules and output MFs. In this work, the optimal number of rules M is equal to the minimal number of clusters used to identify the nonlinear system, as described at the end of Section 3.1. Finally, the adopted fuzzy operators are the product as AND operator, the bounded sum as OR operator, MIN as implication method, and the Centre of Gravity (COG) as defuzzification method.

Section 4 will show the achieved results regarding the fuzzy PI controller parameter tuning using the data sequences from the wind turbine benchmark.

4. Simulation Results

This section describes the simulation results achieved with the method relying on both the fuzzy modelling techniques oriented to the design of the fuzzy controller.

Regarding the proposed control method, the GK clustering algorithm discussed in Section 3.1 with $M = 3$ clusters and a number of shifts $n = 2$ were applied to the estimation and validation sampled data sets $\{P_g(k), \omega_g(k), \beta_r(k)\}$, with $k = 1, 2, \dots, N$ and $N = 440 \times 10^3$. On the other hand, a number of clusters $M = 3$ and $n = 2$ were considered for achieving a suitable clustering of the sampled data sets $\{P_g(k), \omega_g(k), \tau_g(k)\}$. After clustering, the TS model parameters for each output were estimated. Therefore, the i th output $y(t)$ of the wind turbine ($i = 1, \dots, m$ and $m = 2$) continuous-time model of (3) is approximated by a TS fuzzy prototype (13). The relative mean square errors of the output estimations are 0.0254 for the first output and 0.0125 for the second one.

The fitting capabilities of the estimated fuzzy models can be expressed also in terms of the so-called Variance Accounted For (VAF) index [11]. In particular, the VAF value for first output was bigger than 90%, whilst bigger than 99% for the second one. Hence, the fuzzy multiple models seem to approximate the process under investigation quite accurately. As an example, Figure 2 represents simulated output values ω_g and P_g of the TS fuzzy model from the training of the wind turbine benchmark model by using FMID.

Using these identified TS fuzzy prototypes, the model-based approach for determining the fuzzy controller was exploited and applied to the considered wind turbine

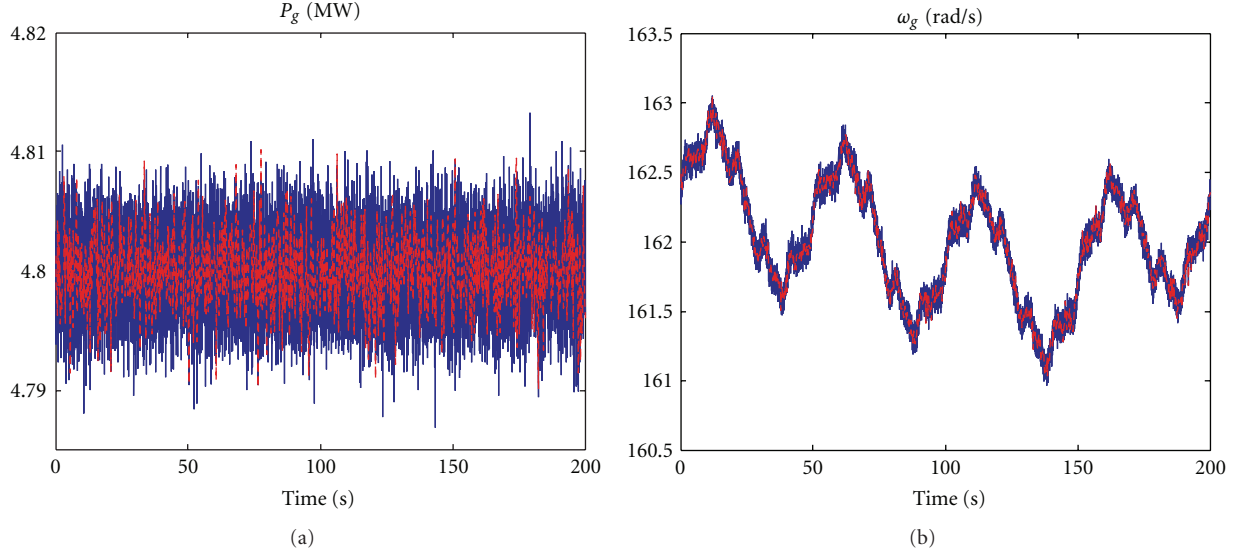


FIGURE 2: TS fuzzy model estimated outputs (red) compared with the measured ones (blue).

benchmark. According to Section 3.2, the parameters of the fuzzy PI controllers were computed.

In the following, the suggested fuzzy PI controllers and the original switching strategy described in Section 2 were implemented in the Matlab and Simulink environments.

The experimental setup employs 2 (Multiple-Input Single-Output) MISO fuzzy PI regulators used for the control of the blade pitch angles and the generator control torque, respectively. As an example, by using the previous (21), the following tuned parameter sets have been computed for the pitch angle control:

$$\begin{aligned} \{K_p^{(1)}, \dots, K_p^{(3)}\} &= \{2.3, 3.1, 3.2\}, \\ \{K_i^{(1)}, \dots, K_i^{(3)}\} &= \{0.2, 0.4, 0.5\}. \end{aligned} \quad (22)$$

In order to compare the advantages of the proposed fuzzy PI strategy, the obtained results were also compared with the ones achieved by using the original switching wind turbine benchmark regulator recalled in Section 2.

The controller capabilities were assessed in simulation by considering different data sequences. In Tables 1 and 2, the per cent Normalised Sum of Squared tracking Error (NSSE) values defined as

$$\text{NSSE\%} = 100 \sqrt{\frac{\sum_{k=1}^N (r(k) - y(k))^2}{\sum_{k=1}^N r^2(k)}} \quad (23)$$

are computed for the designed controllers.

It is worth noting that in *partial load operation* (zone 1), the performance is represented by the comparison between the power produced by the generator, P_g , with respect to the theoretical maximum power output, P_r , given the instant wind speed. On the other hand, in *full load operation* (zone 2), the performance depends on the generator speed, ω_g , with respect to the nominal one, ω_{nom} .

TABLE 1: Controllers in *partial load operation*: NSSE% values.

Data set	Benchmark controller	Fuzzy PI
Estimation data	39.34%	21.36%
Validation data	42.19%	22.17%

TABLE 2: Controllers in *full load operation*: NSSE% values.

Data set	Benchmark controller	Fuzzy PI
Estimation data	19.53%	11.57%
Validation data	21.01%	12.85%

According to these simulation results, the properties of the suggested fuzzy controllers appear better than the original switching regulator described in Section 2.2.

4.1. Robustness and Attraction Domain Evaluation. In this section, further experimental results were reported. They regard the performance evaluation of the developed control scheme with respect to modelling errors and measurement uncertainty. In particular, the simulation of different data sequences was performed by exploiting the wind turbine benchmark simulator and a Matlab Monte-Carlo analysis [21]. In fact, the Monte-Carlo tool is useful at this stage as the control strategy performances depend on the error magnitude due to the model approximation and uncertainty, as well as on input-output measurement errors [22].

In particular, the nonlinear wind turbine simulator originally developed in the Simulink environment [19] was modified by the author in order to vary the statistical properties of the signals used for modelling possible process parameter uncertainty and measurement errors. Therefore, in this case, the Monte-Carlo analysis represents a viable method for analysing some properties of the developed fuzzy

TABLE 3: Simulated wind turbine parameter uncertainties.

Variable	Nominal value	Min. error	Max. error
ρ	1.225 kg/m ³	±0.1%	±20%
A	6647.6 m ²	±0.1%	±20%
J	7.794×10^6 kg/m ²	±0.1%	±30%
p_{gen}	98 s ⁻¹	±0.1%	±20%
N_r	95	±0.1%	±20%
C_p	C_{p0}	±0.1%	±50%
u	u_0	±0.1%	±20%
y	y_0	±0.1%	±20%

TABLE 4: Monte-Carlo analysis for the fuzzy controller: NSSE% values.

Test case	Partial load	Full load
Best case	19.37%	9.57%
Average case	21.19%	11.94%
Worst case	23.19%	13.94%

control scheme, when applied to the considered process. Under this assumption, Table 3 reports the nominal values of the considered wind turbine model parameters with respect to their simulated uncertainty.

The Monte-Carlo analysis was performed by describing these variables as Gaussian stochastic processes, with zero-mean and standard deviations corresponding to minimal and maximal error values in Table 3.

Moreover, it is assumed that the input-output signals u and y and the power coefficient map C_p entries were affected by errors, expressed as per cent standard deviations of the corresponding nominal values u_0 , y_0 , and C_{p0} also reported in Table 3. These values can be motivated by the work from the same author [23], which suggests to provide a polynomial approximation of the map C_p via an identification procedure from the input-output u and y affected by measurement errors.

Therefore, for performance evaluation, reliability and robustness analysis of the fuzzy control scheme, the best, average, and worst values of the NSSE% index were computed, and experimentally evaluated with 500 Monte-Carlo runs [22]. The value of NSSE% is computed for several possible combinations of the parameter values reported in Table 3. Table 4 summarises the results obtained by considering the fuzzy control scheme in Section 3.

In particular, Table 4 summarises the values of the considered performance indices according to the best, worst, and average cases, with reference to the possible combinations of the parameters described in Table 3. Table 4 shows that the proposed fuzzy control scheme allows to maintain good control performances even in the presence of considerable error and uncertainty effects.

The robustness of the controller designs could be evaluated also on the basis of the Domain Of Attraction (DOA) analysis [24]. In fact, the stability of equilibrium points plays a fundamental role in dynamical systems. For

nonlinear dynamical systems an investigation of the stability and robustness properties requires the characterisation of the DOA of an equilibrium point, that is, the set of initial conditions from which the trajectory of the controlled system converges to such a point. In this case, it is clear that estimating or controlling the DOA are very difficult problems because of the complex relationship of this set with the model of the wind turbine. Moreover, as shown in Section 2, the wind turbine system cannot be described as an analytic model. Therefore, the DOA analysis can be performed only in simulation using again the Monte-Carlo tool, which is used for providing random variations of wind turbine initial conditions, in a similar way of Table 3.

Figures 3 and 4 show an example of the results from a Monte-Carlo run, where the wind turbine monitored outputs, that is, the generator power P_g and the generator angular rate ω_g (red) are compared with their reference values (blue), thus assessing the equilibrium point stability, for a perturbed arbitrary initial condition.

In particular, Figure 3 shown that for an arbitrary initial condition randomly changed by the Monte-Carlo tool, the proposed fuzzy TS is able to maintain the stability of the wind turbine. However, the monitor process becomes unstable when controller by the benchmark PI regulator, as highlighted in Figure 4. The simulation results highlight that the benchmark regulator scheme stabilises the wind turbine up to 10% level of variation in its nominal initial conditions, whilst the proposed fuzzy TS control strategy tolerates ranges within 30% of the nominal initial values.

Finally, the results demonstrate that Monte-Carlo simulation is an effective tool for experimentally testing the design robustness, stability, and reliability of the proposed control method with respect to modelling uncertainty and disturbance.

4.2. Comparative Studies. This section provides some comparative results with respect to alternative control approaches, in particular relying on the sliding mode and neural network controllers.

In general, the sliding mode control can be designed on the basis of a linear or a nonlinear model [17]. In both cases, the design procedure is based on the selection of an appropriate switching manifold, and then on the determination of a control law, including a discontinuous term, that ensures the sliding motion in this manifold. However, sliding mode control design for the nonlinear case is generally applied to systems in the so-called *regular form*, which consists of two blocks: one depending on the control, with the same dimension of the control vector, and the other independent. Such regular form may be obtained by means of a nonlinear coordinate transformation. On the other hand, if a linear model is used, the transformation into the regular form and the design of the sliding mode dynamics are simpler, since known results from linear control techniques (i.e., pole placement, eigenstructure assignment, and optimal quadratic) are applicable. This design procedure can be therefore tolerant with respect to uncertainty and modelling errors, since these disturbances are decoupled from the sliding motion.

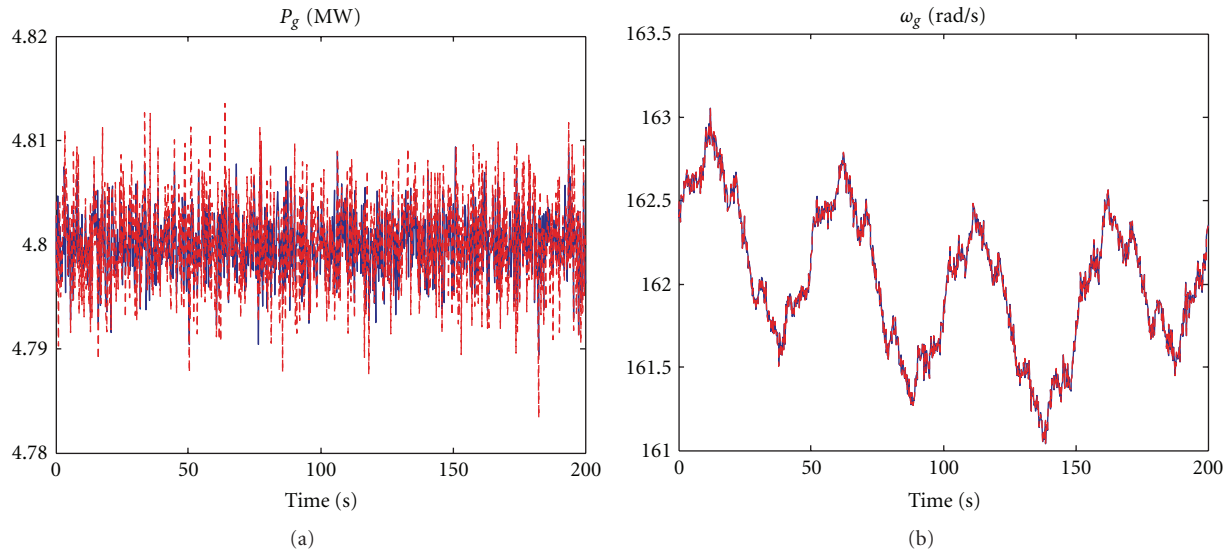


FIGURE 3: P_g and ω_g outputs with the fuzzy TS controller.

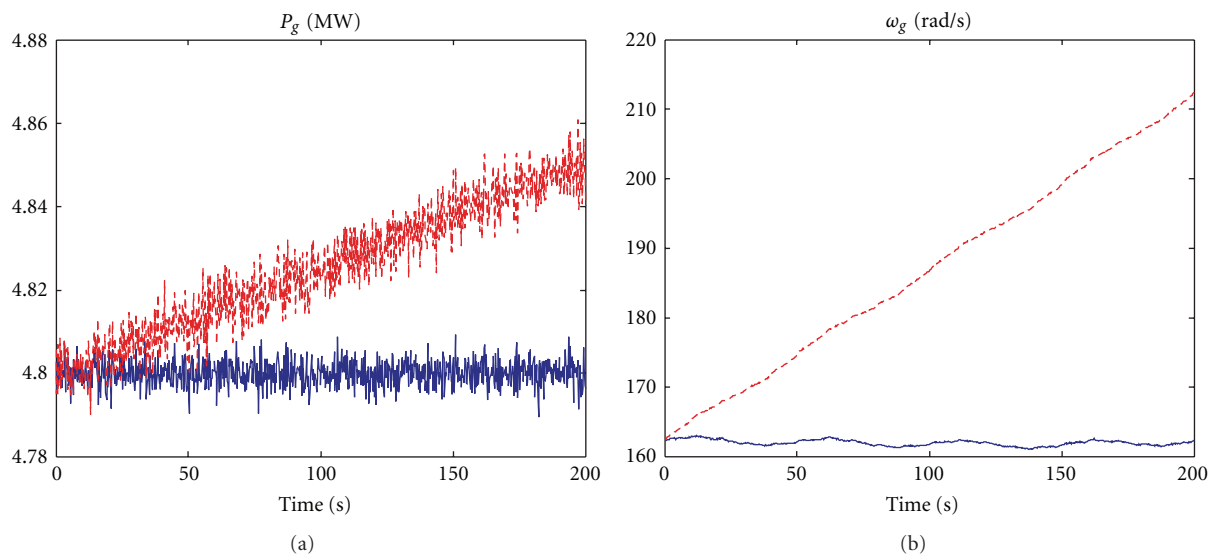


FIGURE 4: P_g and ω_g outputs with the PI benchmark regulator.

The second control scheme exploited for comparison purposes was developed by using the strategy relying on the Neural Network (NN) tool [18]. The NN controller was on a 2-inputs 2-outputs time-delayed three-layer Multi-Layer Perceptron (MLP) NN with 5 neurons in the input layer, 10 neurons in the hidden layer and 2 neurons in the output layer. The NN has been trained in order to provide the optimal reference tracking on the basis of the training patterns and target sequences [18].

In order to provide a brief but clear insight into the above-mentioned control technique, the comparison was performed in the same previous working conditions and based on the NSSE% index suggested at the beginning of Section 4.

It is worth noting that the schemes implemented via the sliding mode or neural controllers do not exploit any adaptation mechanism. In fact, the sliding mode control strategy is able to decouple the uncertainty via the sliding motion, whilst neural networks were designed to *passively tolerate* disturbance and modelling errors.

Tables 5 and 6 summarise the results obtained by comparing the sliding mode and NN techniques analysed in this subsection. It can be seen how the different schemes are able to tolerate uncertainty and errors, thus achieving graceful control performance degradation.

The comparison between Tables 5 and 6 shows that the scheme using the sliding mode controller allows to achieve better performances in terms of tracking error. However,

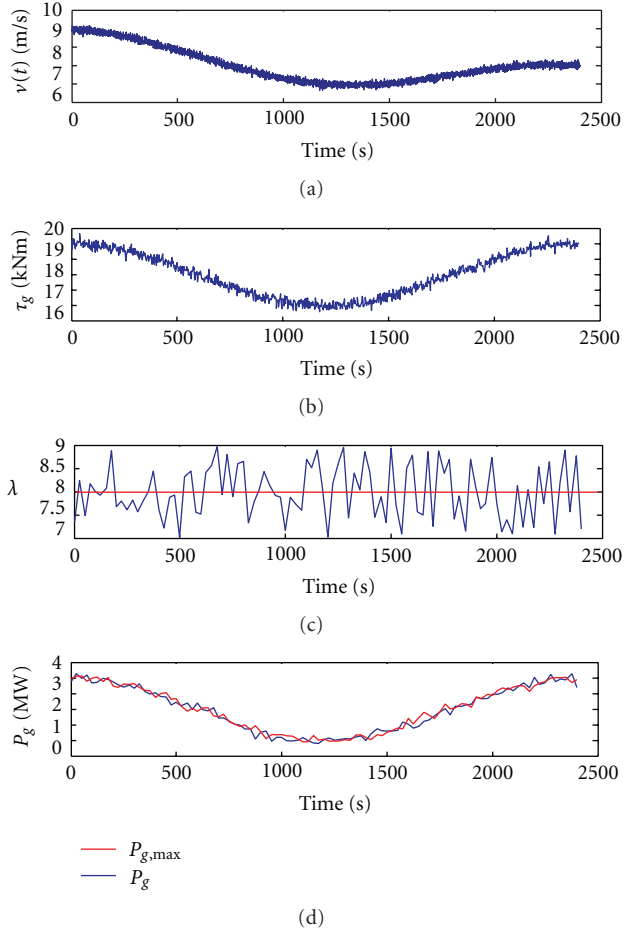


FIGURE 5: Simulations of the wind turbine benchmark working at partial load.

TABLE 5: Monte-Carlo analysis with the sliding mode controller.

Test case	Partial load	Full load
Best case	27.71%	26.55%
Average case	28.52%	27.72%
Worst case	29.19%	28.44%

TABLE 6: Monte-Carlo analysis with the NN controller.

Test case	Partial load	Full load
Best case	36.45%	35.57%
Average case	37.49%	37.94%
Worst case	41.19%	39.94%

the control input energy required by the sliding mode controller is bigger than the other cases. Moreover, the sliding mode controller can increase the computational time considerably with respect to the other solution.

Few further comments can be drawn here. When the modelling of the dynamic system can be perfectly obtained, in general model-based control strategies are preferred. On the other hand, when modelling errors and uncertainty are present, alternative control schemes relying on fuzzy control

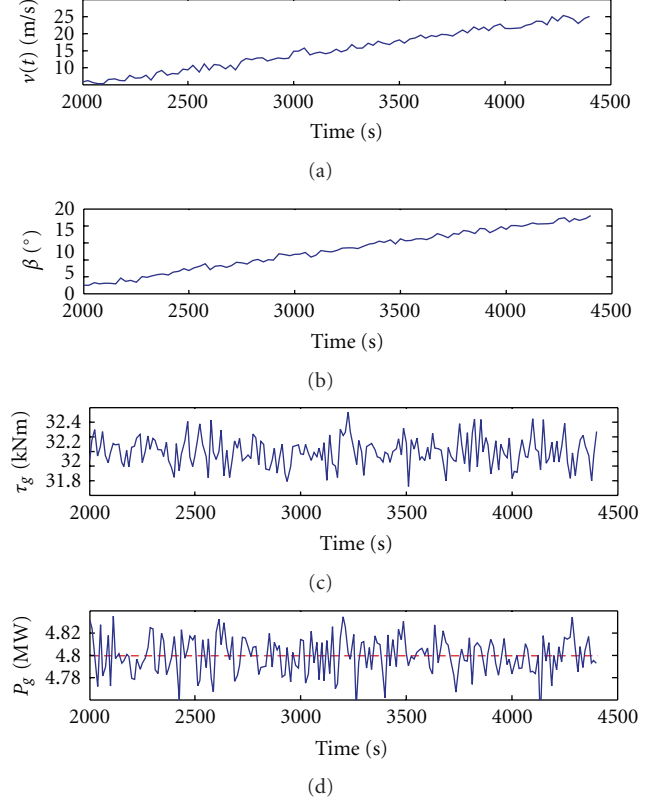


FIGURE 6: Simulations of the wind turbine benchmark operating at full load.

method can show interesting robustness properties in the presence of unmodeled disturbance, modelling mismatch, and measurement errors. With reference to the neural controller, in the case of a controlled system with modelling errors, the offline learning can lead to quite good results. Other explicit disturbance decoupling techniques can take advantage of their robustness capabilities, but with quite complicated and not straightforward design procedures. The NN-based scheme relies on the learning accumulated from offline simulations, but the training stage can be computationally heavy. Regarding the proposed method using the fuzzy tools, it seems rather simple and straightforward, even if optimisation stages can be required.

4.3. Stability Tests. The stability properties of the overall control strategy were checked again by means of a Monte-Carlo campaign based on the wind turbine benchmark simulator when controlled by means of the proposed regulators. In fact, as pointed out in Section 2, the wind turbine system contains the power coefficients map C_p that cannot be described by any analytical model obtained via first principles. Thus, the Monte-Carlo analysis represents the only method for estimating the stability of the developed fuzzy control scheme when applied to the monitored process.

Initial conditions were changed randomly and disturbance affecting the system was simulated during the transient related to the stability analysis.

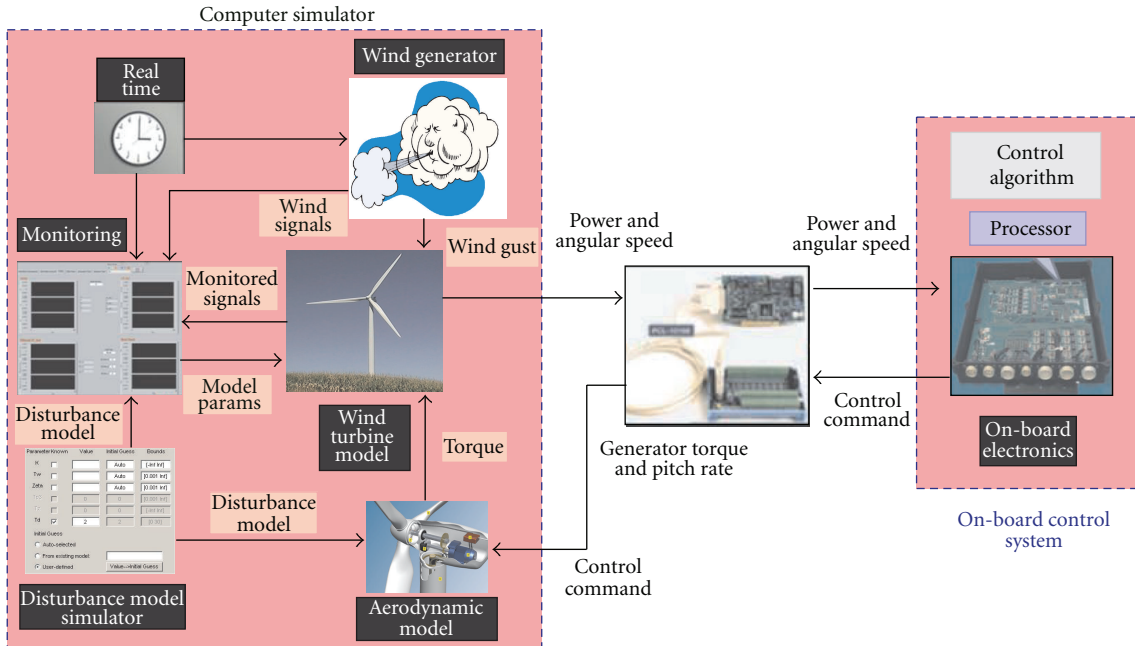


FIGURE 7: Main elements of the HIL test-bed.

All simulations were performed by considering noise signals modelled as band limited white processes, according to the standard deviations reported in Table 3.

As an example of a single Monte-Carlo run, Figure 5 highlighted that the main wind turbine model variables, such as the generator torque τ_g , the tip-speed-ratio λ , and the generator power P_g remain bounded around the reference values, proving the overall system stability in simulation, even in the presence of disturbance and uncertainty. These results refer to the case of partial load operation with the fuzzy controller, when the wind speed $v(t)$ is below 12 m/s.

Figure 5 highlights also that in the first part of the simulation the output power P_g becomes larger than the theoretical one $P_{g,max}$, as the kinetic energy from the rotor shaft is converted into electrical energy produced by the generator. On the other hand, $P_{g,max}$ can be above the generated power, since the inertia of the rotor is accelerated before $P_{g,max}$ can be matched.

As further example of the Monte-Carlo run, the results achieved with the fuzzy controller above rated wind speed are reported in Figure 6.

Figure 6 depicts also the generator speed ω_g and the control input β_r . Also in this situation, the main wind turbine variables remain bounded around the reference values, thus assessing the overall system stability in simulation, even in the presence of modelling errors and noise signals.

It is worth noting that the fuzzy control design followed by the analysis procedures shown in Sections 4.1, 4.2, and 4.3 were developed using the Matlab and Simulink software tools, in order to automate the overall simulation process. These feasibility and reliability studies are of paramount importance for real application of control strategies once implemented to future wind turbine installations.

To this aim, Section 4.4 illustrates how the designed control algorithms are assessed through the hardware in the loop (HIL) test-bed to evaluate their capabilities in a more realistic experimental situation.

4.4. Hardware in the Loop Tests. In order to evaluate the potential of utilising the proposed control algorithms in real applications and investigate their capability to on-board implementation, this section presents the results of the hardware in the loop (HIL) tests. These experimental results serve to validate definitely the desired requirements attributed to the designed algorithms considering almost real conditions that the wind turbine may experiment with during its working situations. For this purpose, HIL test-bed is developed according to Figure 7, which provides the capabilities to validate the developed control algorithms in an almost real-time condition.

This laboratory facility includes the following three elements:

- (i) *Computer simulator*: this simulator, which has been created in the Labview software from the Matlab and Simulink environments, provides the modelling of the wind turbine dynamics considering the factors such as uncertainty, disturbance, measurement errors, and wind turbine component models, as described in Section 2. This software tool allows to monitor the parameters related to the control algorithms and analyse their performance.
- (ii) *On-board electronics*: the control algorithms have been implemented in this element. The electronic device utilised in this module is the AWC 500 system, which in addition satisfies standard wind turbine

TABLE 7: HIL results with the fuzzy controller: NSSE% values.

Partial load	Full load
22.81%	12.83%

technical specifications. This element also provides the flexibility to implement and evaluate the different control algorithms. As it can be seen from Figure 7, the on-board electronics receives the wind turbine power and generator angular rate as its inputs, and after the algorithm processing, generates the generator torque and pitch command output signals transmitted to the computer simulator. The generator torque and the pitch commands are generated by the proposed control algorithms and are applied to the wind turbine model simulator to guarantee its stability and the required specifications.

- (iii) *Interface circuits*: they consist of the appropriately selected input-output cards, which receive the output signals from the computer simulator and transmit the output signals generated by the control algorithms.

The results achieved from one test are summarised in Table 7 for the proposed fuzzy controller.

Table 7 illustrates that there are some deviations between the achieved results, but consistent with the ones of Table 4 from the Monte-Carlo analysis. In fact, the performances in the simulation case are somehow better than the HIL experimental case, which is reasonable due to the issues detailed below.

- (1) Accuracy of the float calculations in the on-board electronics processor is more restrictive than the CPU of the computer simulator.
- (2) The major deviation between the results originates from the analog to digital and digital to analog conversions.

Since the data must be transferred between the on-board electronics and the computer simulator, a 16 bit conversion is inevitable, so this conversion error may lead to the deterioration of the experimental results. Note that, since the real situations do not need to transfer data between the on-board electronics and the computer, this error is not a problem and is consistent with the results already achieved via the Monte-Carlo tool. Moreover, although there are some deviations between the simulation and the experimental results, but due to the reasons stated above, these deviations are not critical, and the results obtained are accurate enough for real wind turbine applications.

5. Conclusion

The paper is focused on the control design problem for wind turbines, since they are complex nonlinear dynamic systems. Moreover, their aerodynamics are nonlinear, unsteady, and complex, and their modelling is thus challenging. Therefore, the design of control algorithms for wind turbines must

account for these complexities. This paper showed how a fuzzy control design method can be suggested to improve classic control limitations, such as PI standard regulators. Therefore, the application example of a viable, simple, and straightforward control design was provided, through extensive simulations of a wind turbine prototype. Tests on the considered benchmark process and Monte-Carlo analysis represented the tools for assessing experimentally the properties of the proposed fuzzy control scheme, in the presence of modelling and measurement errors. The developed control method was also compared with other different approaches. To highlight the potential of the proposed control algorithms in real applications, hardware in the loop test facility was planned to study the digital implementation of the designed strategies. The test results showed again that the developed schemes maintained their desired performances, which validated their feasibility also in real-time implementations. Future works aim to perform sustainable and dependability analyses that are of paramount importance for real wind turbine applications.

References

- [1] K. E. Johnson, L. Y. Pao, M. J. Balas, and L. J. Fingersh, "Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 70–81, 2006.
- [2] P. F. Odgaard, C. Damgaard, and R. Nielsen, "On-line estimation of wind turbine power coefficients using unknown input observers," in *Proceedings of the 17th World Congress, International Federation of Automatic Control (IFAC '08)*, Seoul, Korea, July 2008.
- [3] Z. Hameeda, Y. Honga, Y. Choa, S. Ahnb, and C. Song, "Condition monitoring and fault detection of wind turbines and related algorithms: a review," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 1, pp. 1–39, 2009.
- [4] X. F. Zhang, D. P. Xu, and Y. B. Liu, "Adaptive optimal fuzzy control for variable speed fixed pitch wind turbines," in *Proceedings of the 5th World Congress on Intelligent Control and Automation, Conference Proceedings (WCICA '04)*, pp. 2481–2485, June 2004.
- [5] R. Sakamoto, T. Senjyu, T. Kinjo, N. Urasaki, and T. Funabashi, "Output power leveling of wind turbine generator by pitch angle control using adaptive control method," in *Proceedings of the International Conference on Power System Technology (POWERCON '04)*, pp. 834–839, November 2004.
- [6] E. A. Bossanyi, "Developments in closed loop controller design for wind turbines," in *Proceedings of the 19th ASME Wind Energy Conference (ASME '00)*, pp. 64–74, ASME, Reno, Nev, USA, 2000.
- [7] L. J. Fingersh and K. E. Johnson, "Baseline results and future plans for the NREL controls advanced research turbine," in *Proceedings of the 23rd ASME Wind Energy Conference*, pp. 87–93, ASME, Reno, Nev, USA, January 2004.
- [8] K. E. Johnson, L. J. Fingersh, M. J. Balas, and L. Y. Pao, "Methods for increasing region 2 power capture on a variable-speed wind turbine," *Journal of Solar Energy Engineering*, vol. 126, no. 4, pp. 1092–1100, 2004.
- [9] M. Geyler and P. Caselitz, "Individual blade pitch control design for load reduction on wind turbines," in *Proceedings of the European Wind Energy Conference and Exhibition*, pp. 7–10, Milan, Italy, May 2007.

- [10] A. D. Wright, L. J. Fingersh, and M. J. Balas, "Testing state-space controls for the controls advanced research turbine," *Journal of Solar Energy Engineering*, vol. 128, no. 4, pp. 506–515, 2006.
- [11] R. Babuška, *Fuzzy Modeling for Control*, Kluwer Academic Publishers, 1998.
- [12] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [13] S. Simani, C. Fantuzzi, R. Rovatti, and S. Beghelli, "Noise rejection in parameters identification for piecewise linear fuzzy models," in *Proceedings of the IEEE International Conference on Fuzzy Systems Part 2 (of 2)*, pp. 378–383, May 1998.
- [14] S. Simani, C. Fantuzzi, R. Rovatti, and S. Beghelli, "Parameter identification for piecewise-affine fuzzy models in noisy environment," *International Journal of Approximate Reasoning*, vol. 22, no. 1, pp. 149–167, 1999.
- [15] V. Bobál, J. Böhm, J. Fessl, and J. Macháček, *Digital Self-Tuning Controllers: Algorithms, Implementation and Applications. Advanced Textbooks in Control and Signal Processing*, Springer, 1st edition, 2005.
- [16] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault tolerant control of wind turbines—a benchmark model," in *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, vol. 1, pp. 155–160, Barcelona, Spain, July 2009.
- [17] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory and Applications*, Taylor & Francis, London, UK, 1st edition, 1998.
- [18] J. Korbicz, J. M. Koscielny, Z. Kowalczyk, and W. Cholewa, Eds., *Fault Diagnosis: Models, Artificial Intelligence, Applications*, Springer, 1st edition, 2004.
- [19] P. F. Odgaard and J. Stoustrup, "Unknown input observer based scheme for detecting faults in a wind turbine converter," in *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, vol. 1, Elsevier, Barcelona, Spain, June 2009.
- [20] R. Babuška, *Fuzzy Modelling and Identification Toolbox*, Control Engineering Laboratory, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, 2000.
- [21] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science*, Springer, New York, NY, USA, 2001.
- [22] D. Lieber, A. Nemirovskii, and R. Y. Rubinstein, "A fast Monte Carlo method for evaluating reliability indexes," *IEEE Transactions on Reliability*, vol. 48, no. 3, pp. 256–261, 1999.
- [23] S. Simani and P. Castaldi, "Estimation of the power coefficient map for a wind turbine system," in *Proceedings of the 9th European Workshop on Advanced Control and Diagnosis (ACD '11)*, Budapest University of Technology and Economics, Ed., no. paper 13, pp. 1–7, MTA SZTAKI Computer and Automation Research Institute, Hungarian Academy of Sciences, and BME Budapest University of Technology and Economics, BME Budapest University of Technology and Economics, Budapest, Hungary, November 2011.
- [24] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, 3rd edition, 2001.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

