Data-Driven and Model-Based Control Techniques for a Wind Turbine Benchmark Model

Silvio Simani[1,*], Saverio Farsoni[1], and Paolo Castaldi[2]

[1]Department of Engineering, Ferrara University, Ferrara, FE, 44122 Italy

[2]Department of Electrics and Informatics, Bologna University, Bologna, BO, 40121 Italy

Abstract: Wind turbine plants are complex dynamic and uncertain processes driven by stochastic inputs and disturbances, as well as different loads represented by gyroscopic, centrifugal, and gravitational forces. Moreover, as their aerodynamic models are nonlinear, both modelling and control become challenging problems. On one hand, high–fidelity simulators should contain different parameters and variables in order to accurately describe the main dynamic system behaviour. Therefore, the development of modelling and control for wind turbine systems should consider these complexity aspects. On the other hand, these control solutions have to include the main wind turbine dynamic characteristics without becoming too complicated. The main point of this paper is thus to provide two practical examples of development of robust control strategies when applied to a simulated wind turbine plant. Extended simulations with the wind turbine benchhmark model and the Monte–Carlo tool represent the instruments for assessing the robustness and reliability aspects of the developed control methodologies when the model–reality mismatch and measurement errors are also considered. Advantages and drawbacks of these regulation methods are also highlighted with respect to different control strategies via proper performance metrics.

Keywords: Wind turbine simulator, Data–driven and model-based approaches, Fuzzy identification, On-line estimation, Robustness and reliability.

1. INTRODUCTION

Wind turbine plants represent complex and nonlinear dynamic systems usually driven by stochastic inputs and different disturbances describing gravitational, centrifugal, and gyroscopic loads. Moreover, their aerodynamic models are uncertain and nonlinear, whilst wind turbine rotors are subject to complex turbulent wind fields, especially in large systems, thus yielding to extreme fatigue loading conditions. In this way, the development of viable, robust and reliable control solutions for wind turbines can become a challenging issue [10].

Usually, a model–based control design requires an accurate description of the system under investigation, which has to include different parameters and variables in order to model the most important nonlinear and dynamic aspects. Moreover, the wind turbine working conditions can produce further problems to the design of the control method. In general, commercial codes are not able to adequately describe the wind turbine overall dynamic behaviour; usually, special simulation software solutions are used. On the other hand, control schemes have to manage the most important turbine dynamics, without being too complex and unwieldy. Control methods for wind turbines usually rely on the signals from sensors and actuators, with a system that connects these elements together. Hardware or software modules elaborate these signals to generate the output signals for actuators. The main feature of the control law consists of maintaining safe and reliable working conditions of the wind turbine, while achieving prescribed control performances, and allowing for optimal energy conversion, as shown *e.g.* in recent works applied to the same wind turbine model considered in this work [16].

Today's wind turbines can implement several control strategies to allow for the required performances. Some turbines use passive control methods, such as in fixed–pitch, stall control machines. In this case, the system is designed so that the power is limited above rated wind speed through the blade stall. Therefore, the control of the blades is not required [10]. In this case, the rotational speed control is proposed thus avoiding the inaccuracy of measuring the wind speed. Rotors with pitch regulation are usually used for constant–speed plants, in order to provide a power control that works better than the blade stall solution. In these machines, the blade pitching is controlled in order to provide optimal power conversion with respect to modelling errors, wind gusts and disturbance. However, when the system works at constant speed and below rated wind speed, the optimal conversion rate cannot be obtained. Therefore, in order to maximise the power conversion rate, the rotational speed of the turbine must vary with wind

*Address correspondence to this author at the Department of Engineering, Ferrara University, Ferrara, FE, 44122 Italy; Tel: +390532974844;
E-mail: silvio.simani@unife.it

E-ISSN: 2409-9694/22

speed. Blade pitch control is thus used also above the rated wind speed [10] . A different control method can introduce the yaw regulation to orient the machine into the wind field. A yaw error reference from a nacelle–mounted wind direction sensor system must be included in order to calculate this reference signal [32].

Regarding the regulation strategies proposed in this paper, two control design examples are described and applied to a wind turbine system. The wind turbine model exploited in this work is freely available for the Matlab and Simulink environments, and already proposed as benchmark for an international competition regarding the validation of fault diagnosis and fault tolerant control approaches [16].

In particular, a first *data–driven* method relying on a fuzzy identification approach to the control design is considered. In fact, since the wind turbine mathematical model is nonlinear with uncertain inputs, fuzzy modelling represents an alternative tool for obtaining the mathematical description of the controlled process. In contrast to purely nonlinear identification schemes, see *e.g.* [11], fuzzy modelling and identification methods are able to directly provide nonlinear models from the measured input–output signals. Therefore, this paper suggests to model the wind turbine plant via Takagi–Sugeno (TS) fuzzy prototypes [1], whose parameters are obtained by identification procedures. This approach is also motivated by previous works by the same authors [24] . On the other hand, concerning the control design, the paper proposes also a fuzzy control method for the regulation of the blade pitch angle, and the generator torque of the wind turbine system.

With respect to similar works, see *e.g.* [7], this paper suggests an off–line identification approach, without any on–line optimisation schemes, thus enhancing real–time implementations. Note also that the works by the same authors, see *e.g.* [23], addressed a different design procedure of the fuzzy regulator, that consists of fuzzy PI controllers. On the other hand, this paper proposes the direct estimation of the fuzzy regulator by means of an *identification* scheme.

Regarding the second *model–based* strategy presented in this paper, it relies on an adaptive control scheme [12]. Again, with respect to pure nonlinear control methods [20], it does not require a detailed knowledge about the model structure. Therefore, this work suggests the implementation of controllers based on adaptive schemes, used for the recursive derivation of the controller model.

In particular, a recursive Frisch scheme extended to the adaptive case for control design is considered in this study, as proposed *e.g.* in [23] by the same authors, which makes use of exponential forgetting laws. This allows the on–line application of the Frisch scheme to derive the parameters of a time–varying controller.

Since it is necessary to evaluate the robustness and the reliability of the designed control methods with respect to modelling uncertainties, disturbance, and measurement errors, the verification and validation tools use extensive Monte–Carlo simulations. In fact, the wind turbine system contains elements that cannot be described by analytical models. Thus, the Monte–Carlo analysis represents a solution for testing the robustness and reliability features of the control schemes when applied to the wind turbine model. This paper compares the proposed methodologies also with respect to different control methods based on sliding mode techniques, neural controllers, or gain scheduling methods. However, with respect again to [23] by the same authors, different comparisons are proposed in this work that exploit proper performance metrics.

It is worth noting the main contributions brought by this paper. The first one proposes a methodology relying on a fuzzy inference system that requires the clustering of the available data into subsets characterised by linear behaviours. The integration between clusters and linear regression is exploited, thus allowing for the combination of fuzzy logic techniques with system identification methodologies. This study proposes the use of fuzzy models in the form of Takagi–Sugeno prototypes as they are able to describe nonlinear dynamic relations with arbitrary accuracy. The switching between the local affine submodels is achieved through a smooth function of the system state defined exploiting the fuzzy set theory and its tools. The main advantage of this approach is motivated by the availability of proper simulation tools already present in the Matlab and Simulink environments. Moreover, this approach has shown very powerful features, since allowed several degrees of freedom that can be exploited for improving the accuracy of the achieved model. On the other hand, the computational complexity increases when extensive optimisation procedures are required for obtaining the optimal structure of the fuzzy models.

However, this aspect is not critical as this optimisation phase can be performed off–line.

Regarding the second solution exploited for obtaining the mathematical description of the controller, it relies on a recursive algorithm applied to a well–established description of the system under investigation as error–in–variable model. The main advantage of the methodology is the optimal filtering of the uncertainty and disturbance effects, without any preliminary assumtpion on the noise process. On one hand, the procedure can be easily implemented on–line by means of the simulation tools already available in the Matlab and Simulink. On the other hand, specific designs are required for the transformation of the adaptive algorithms into iterative schemes that may be simulated in real–time, without increasing the computational complexity of the coding solution.

Finally, this work is organised as follows. Section 2 recalls the wind turbine model considered for control design purposes. Section 3 addresses the data–driven scheme exploited for the derivation of the fuzzy controller, proposed in Section 3.1. On the other hand, the model–based control design is considered in Section 3.2, based on its mathematical derivation also described in Section 3. The achieved results and comparisons with different control strategies are outlined in Section 4. The robustness and reliability features of the developed control strategies are also investigated. Finally, Section 5 ends the paper by summarising the main achievements of the work.

2. WIND TURBINE BENCHMARK MODEL AND ITS GOVERNOR

This section outlines the wind turbine model, whose sampled inputs and outputs will be used for the proposed control designs, as shown in Section 3. In particular, Section 2.1 recalls the wind turbine model, whilst Section 2.2 summarises its baseline regulator, as proposed in [17].

2.1. Wind Turbine Model

The wind turbine system exploited in this work uses a nonlinear dynamic model representing the wind acting on the wind turbine blades, thus producing the movement of the low–speed rotor shaft. The higher speed required by the electric converter is produced by means of a gear box. The simulator is described in more detail *e.g.* in [17, 15]. A scheme of the wind turbine simulator considered in this paper is represented in Figure **1** with its main blocks.

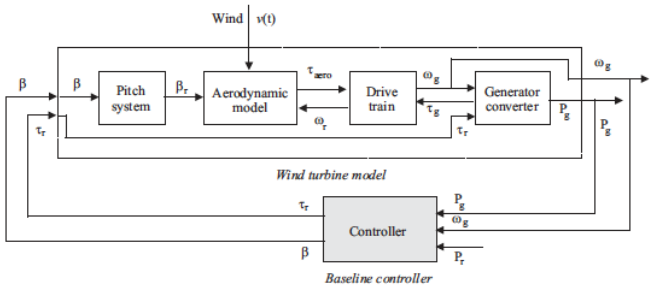

Figure 1: Scheme of the wind turbine simulator.

Both the generator speed ω_g and the generator power P_g are controller by means of the 2 control inputs representing the generator torque $\tau_g(t)$ and the blade pitch angle $\beta(t)$. Several signals can be acquired from the wind turbine simulator. In particular, the main signals available from the wind turbine simulator and used for control purpose are listed in the following.

- β : blade pitch angle [deg];
- ω_g : generator/converter speed [rad/s];
- P_g : generator power P_g [W];
- ω_r : rotor speed [rad/s];
- τ_r : requested torque [N m];
- τ_g : generator torque [N m];
- τ_{aero} : aerodynamic torque [N m];
- v : wind speed [m/s];

The aerodynamic model defining the aerodynamic torque provides the $\tau_{aero}(t)$ signal, which is a nonlinear function of the wind speed $v(t)$. This measurement is very difficult to be acquired correctly, as described in [15].

The aerodynamic model reported in Figure **1** is described by the following relation:

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

where the variable ρ represents the air density, whilst A is the effective rotor area.

The pitch system implemented in the wind turbine benchmark as depicted in Figure **1** uses a second–order transfer function, represented by Eq. (2) :

$$\frac{\beta_r(s)}{\beta(s)} = \frac{\omega_n^2}{s^2 + 2\chi\omega_n + \omega_n^2} \tag{2}$$

where β_r is the actuated pitch angle, which is exploited in Eq. (1) , whilst β is the controlled pitch angle. ω_n is the natural frequency of the pitch actuator model, and χ its damping ratio.

Another important variable is represented by the so–called tip–speed ratio, which is defined as:

$$\lambda(t) = \frac{\omega_r(t)R}{v(t)} \tag{3}$$

with R the rotor radius. $C_p(\cdot)$ represents the power coefficient, that is normally represented via a two–dimensional map [15] . The expression of Eq. (1) allows the computation of the signal $\tau_{aero}(t)$ (highlighted in Figure **1**) by means of the estimated wind speed $v(t)$, and the signals $\beta(t)$ 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 justifies the robust approaches described in Section 3. Moreover, the nonlinearity represented by the expressions of Eqs. (1) and (3) motivates the required reliable and robust control approaches suggested in this work.

The wind turbine simulator includes a two–mass model that is exploited to describe the drive–train system depicted in Figure **1**, as shown by the following linear state–space representation [17]:

$$\begin{bmatrix} \dot{\omega}_r(t) \\ \dot{\omega}_g(t) \\ \dot{\theta}_\Delta(t) \end{bmatrix} = A_{dt} \begin{bmatrix} \omega_r(t) \\ \omega_g(t) \\ \theta_\Delta(t) \end{bmatrix} + B_{dt} \begin{bmatrix} \tau_{aero}(t) \\ \tau_g(t) \end{bmatrix} \tag{4}$$

where the matrices A_{dt} and B_{dt} are defined as:

$$A_{dt} = \begin{bmatrix} -\frac{B_{dt} - B_r}{J_r} & \frac{B_{dt}}{N_g J_r} & -\frac{K_{dt}}{J_r} \\ & -\frac{\eta_{dt} B_{dt}}{N_g^2} - B_g & \\ \frac{\eta_{dt} B_{dt}}{N_g J_g} & \frac{}{J_g} & \frac{\eta_{dt} K_{dt}}{N_g J_g} \end{bmatrix} \tag{5}$$

and:

$$B_{dt} = \begin{bmatrix} -\frac{1}{J_r} & 0 \\ 0 & -\frac{1}{J_g} \end{bmatrix} \tag{6}$$

where J_r is the momentum of inertia of rotor shaft, K_{dt} is the torsion stiffness of the drive–train, B_{dt} is the torsion damping coefficient of the drive–train, B_g is the viscous friction of the generator shaft, N_g is the gear ratio, J_g is the moment of inertia of the generator shaft, η_{dt} is the efficiency of the drive–train, and θ_Δ is the torsion angle of the drive train. Note that the benchmark simulator considered in this work does not include possible nonlinear gearbox dynamics, as addressed *e.g.* in [21, 28, 29] . However, the data–driven approach proposed in this study could be able to include this further nonlinearity for the design of the control solutions.

Moreover, the generator/converter dynamics are described as a first–order transfer function, as highlighted by Eq. (7):

$$\frac{\tau_g(s)}{\tau_r(s)} = \frac{\alpha_{gc}}{s + \alpha_{gc}} \tag{7}$$

s being the Laplace operator, $1/\alpha_{gc}$ is the time constant of the generator/converter, whilst the power P_g produced by the generator is given by Eq. (8):

$$P_g(t) = \eta_g \omega_g \tau_g(t) \tag{8}$$

with η_g denoting the efficiency of the generator. More details regarding the considered simulator are in [15] .

Under these assumptions, the complete state–space description of the wind turbine model has the form of Eq. (9):

$$\begin{cases} \dot{x}_c(t) &= f_c\left(x_c(t), u(t)\right) \\ y(t) &= x_c(t) \end{cases} \tag{9}$$

where $u(t) = \left[\beta(t), \tau_g(t)\right]^T$, $y(t) = \left[P_g(t), \omega_g(t)\right]^T$, and $x_c(t) = \left[P_g(t), \omega_g(t)\right]^T$ are the control inputs, the monitored output measurements, and the state vector, respectively, as shown in Figure **1**. $P_g(t)$ is the generator power measurement, whilst $f_c(\cdot)$ represents the continuous–time nonlinear function that will be approximated via discrete–time models from N sampled data $u(t_k)$ and $y(t_k)$, with the sample index $k = 1, 2, \ldots N$ and sampling time T_s, such that $t_k = kT_s$, as presented in Section 3. The model parameters, and the map $C_p(\beta, \lambda)$ are chosen in order to represent a realistic wind turbine plant [15] .

Table 1: Sensor Standard Deviation Values used in the Wind Turbine Simulator

Variable	$v(t)$	ω_r	ω_g
Std. Dev. Value	0.5 m/s	0.025 rad/s	0.05 rad/s
Variable	τ_g	P_g	β
Std. Dev. Value	90 Nm	10^3 W	0.2 deg

Moreover, the input and output measurements available from the wind turbine simulator are assumed to be acquired via sensors that introduce additive Gaussian noise processes with zero mean and standard deviation values summarised in Table **1**.

On the other hand, this benchmark model set–up uses a predefined wind speed sequence $v(t)$ consisting of real measured wind data of a wind park from 0 to 4400 s. Figure **2** highlights that this wind speed covers the range from 5 to 20 m/s, with a few spikes at 25 m/s, which is a good coverage of normal operational for a wind turbine.

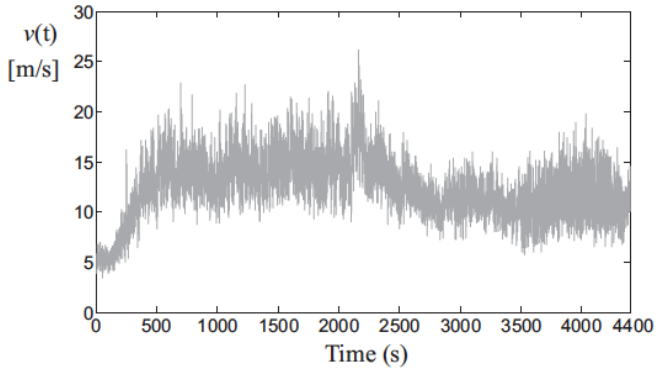

Figure 2: The wind speed sequence $v(t)$ used in the benchmark model.

It is worth noting also that the nonlinearity represented by Eqs. (1) and (3) is sketched in Figure **3**, for different values of $\lambda(t)$ (*i.e.* $v(t)$) and $\beta(t)$.

2.2. Wind Turbine Baseline Regulator

With reference to the baseline control system for the wind tubine model, Figure **3** also shows the power curve that highlights the the so–called partial load (region 1) and full load (region 2) working conditions of a wind turbine [17]. In fact, the baseline controller implemented in the wind turbine simulator works in these 2 operating conditions. A schematic diagram of the baseline wind turbine controller system is detailed in Figure **4**.

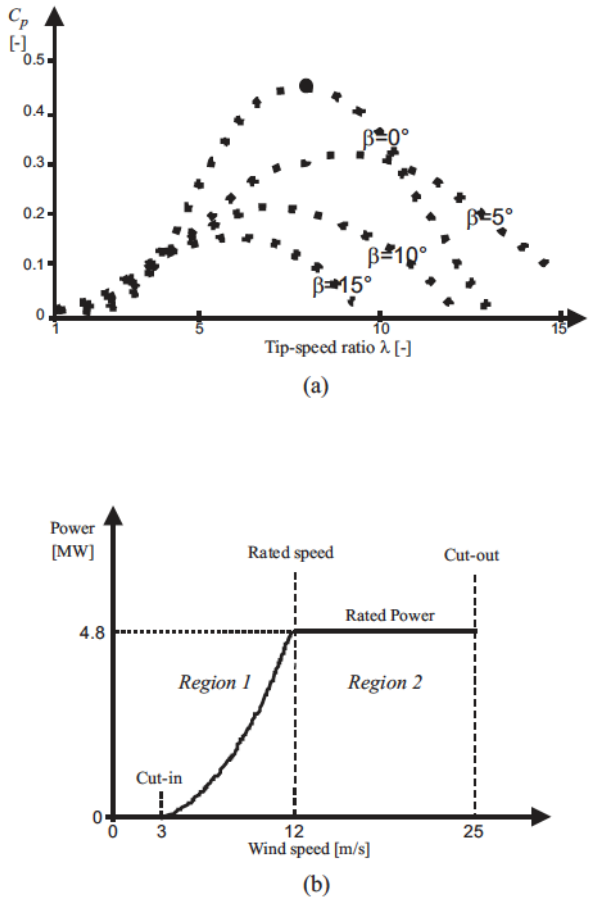

Figure 3: Examples of (**a**) power coefficient function and (**b**) generated power (**b**).

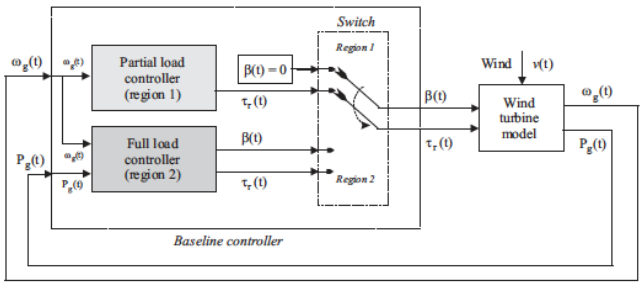

Figure 4: The details of the baseline wind turbine controller.

In particular, Figure **4** highlights that in partial load working condition, the optimal tracking is achieved without any pitching of the blades, which are fixed to 0 degrees. In this case, λ is constant at its optimal value λ_{opt}, that is defined by the maximal value of the power coefficient map C_p when $\beta = 0$ degrees, as shown in Figure **3**(**a**). Therefore, this working condition is completely defined by setting $\tau_g = \tau_r$ (*i.e.* the generator torque is equal to the required reference value) with pitch angle $\beta = 0$ degrees. Note that this assumption is no longer valid when the wind turbine is working in full load conditions, when the signal $\beta_r \sum_{i=1}^{n} (X_i - \overline{X})^2$ is determined by the controller.

The reference torque signal τ_g shown in Figure **1** is computed as:

$$\tau_r = K_{opt} \omega_g^2 \tag{10}$$

where:

$$K_{opt} = \frac{1}{2} \rho A R^3 \frac{C_{p_{max}}}{\lambda_{opt}^3} \tag{11}$$

with $C_{p_{max}}$ the maximal value of C_p, related the to λ_{opt}, *i.e.* the optimal tip–speed ratio, as sketched in Figure **3**(**a**).

When the power reference $P_r = 4.8MW$ is achieved by the wind turbine system (it corresponds to the so–called rated power) [17] , and the wind speed $v(t)$ increases, the controller is switched to the control region 2 (full load condition). In this working condition (region 2), the control objective consists of tracking the power reference P_r, obtained by regulating β, such that the C_p is decreased, as shown in Figure **3**(**a**). In a traditional industrial control scheme, usually a PI controller is used to keep ω_g at the prescribed value by changing β; the second input of the controller is τ_g.

The baseline controller considered in this work was implemented with a sample frequency at 100 Hz, *i.e.* $T_s = 0.01$ s. In full load conditions, *i.e.* in region 2, the actuated input β is controlled via the relations of Eq. (12) [15] :

$$\begin{cases} \beta(t_k) &= \beta(t_{k-1}) + k_p e(t_k) + \\ & \quad + \left(k_i T_s - k_p\right) e(t_{k-1}) \\ e(t_k) &= \omega_g(t_k) - \omega_{nom} \end{cases} \tag{12}$$

with the sample index $k = 1,2,\ldots,N$. The parameters for this PI speed controller are $k_i = 0.5$ [1/s] and $k_p = 3$ [deg/rad/s] [15] . For the case of the wind turbine system considered in this paper, the constant ω_{nom} is equal to 162 [rad/s] [17] .

The control of the further input τ_g shown in Figure **1**, a second PI regulator is used, in the form of Eq. (13):

$$\begin{cases} \tau_r(t_k) &= \tau_r(t_{k-1}) + k_p e(t_k) + \\ & \quad + \left(k_i T_s - k_p\right) e(t_{k-1}) \\ e(t_k) &= P_g(t_k) - P_r \end{cases} \tag{13}$$

The parameters for this second PI power controller are $k_i = 0.014$ and $k_p = 447 \times 10^{-6}$ [15].

Finally, note that in region 1 (partial load, below the rated wind speed) the wind turbine is regulated only by means of the torque input $\tau_g(t)$. In this situation, the blade pitching system is not exploited to achieve the optimal power conversion, as highlighted in Figure **4**. On the other hand, in region 2 (full load, above the rated wind speed), the wind turbine control regulates both the blade pitch angle $\beta(t)$ and the control torque $\tau_g(t)$. The wind turbine Simulink model considered in this work includes also saturation blocks that limit the values of the control signals, which were not reported in Figures **1** and **4**.

3. DATA–DRIVEN AND MODEL–BASED CONTROL TECHNIQUES

This section describes the two approaches considered in this paper for obtaining the control laws by using data–driven and model–based methodologies. Once a suitable mathematical description of the monitored process is provided, the derivation of the controller structure is sketched in Section 3.1 for the fuzzy approach, whilst Section 3.2 proposes a different method relying on an adaptive technique.

3.1. Fuzzy Control Strategy

The first method proposed in this paper for the derivation of the wind turbine controller is based on a fuzzy clustering technique to partition the available data into subsets characterised by linear behaviours. The integration between clusters and linear regression is exploited, thus allowing for the combination of fuzzy logic techniques with system identification methodologies. These tools are already available and implemented in the Matlab Fuzzy Modelling and

IDentification (FMID) Toolbox recalled below [1]. This study proposes the use of TS fuzzy prototypes since they are able to model nonlinear dynamic systems with arbitrary accuracy [1]. The switching between the local affine submodels is achieved through a smooth function of the system state defined exploiting the fuzzy set theory and its tools.

In more detail, the fuzzy estimation scheme relies on a two–step algorithm, in which, the working regions are first defined by exploiting the data fuzzy clustering tool, *i.e.* the Gustafson–Kessel (GK) method [1]. On the other hand, the second step performs the identification of the controller structure and its parameters using the estimation method proposed by the same authors in [24]. This estimation approach can be considered as a generalisation of the general least–squares method for hybrid models.

Under these assumptions, the TS fuzzy prototypes have the form of the model of Eq. (14):

$$y(t_{k+1}) = \frac{\sum_{i=1}^{M} \mu_i\left(\boldsymbol{x}(t_k)\right) y_i(t_k)}{\sum_{i=1}^{M} \mu_i\left(\boldsymbol{x}(t_k)\right)} \tag{14}$$

where $y_i(t_k) = \boldsymbol{a}_i^T \boldsymbol{x}(t_k) + b_i$, with $\boldsymbol{a}_i$ the parameter vector (regressand), and b_i is the scalar offset. $\boldsymbol{x}(t_k)$ represents the regressor vector containing the delayed samples of the signals $u(t_k)$ and $y(t_k)$.

Note that the discrete–time description of Eq. (14), after the proper estimation of the parameter vector [1], will be used for reconstructing the sampled outputs $P_g(t)$ and $\omega_g(t)$ of the continuous–time nonlinear model of Eq. (9) fed by the sampled input signals $\beta(t)$ and $\tau_g(t)$.

The antecedent fuzzy sets μ_i that determine the switching among the different submodels i in Eq. (14) are estimated using the input–output data acquired from the wind turbine simulator, *i.e.* the input and output sampled signals $\beta(t)$, $\tau_g(t)$, and $\omega_g(t)$ and $P_g(t)$. These data are organised into proper clusters where affine relations hold, as described in [1]. The consequent parameters $\boldsymbol{a}_i$ and b_i are also identified from these input–output data by means of the estimation methodology proposed in [24] . This identification scheme exploited for the estimation of the TS model parameters has been integrated into the FMID toolbox for Matlab by the authors. This approach is preferable when the TS model of Eq. (14) is used as predictor, since it derives the consequent parameters via the so–called Frisch scheme, developed for the Errors–In–Variables (EIV) structures [24].

Once the description of the monitored process is obtained in the form of Eq. (14) , the data–driven approach for the design of the fuzzy controller is exploited. As already remarked, this design procedure differs from the approach proposed in [22] . In fact, the control design proposed in this paper relies on the so–called *inverse model* principle, which is solved using the fuzzy identification approach recalled above.

Note that as explained in the following, the fuzzy methodology is exploited twice. First, the fuzzy modelling and identification approach is used to derive a fuzzy representation of the process under investigation, by means of its sampled inputs and outputs. Second, by means of this fuzzy model, and in particular using its state, the design of the fuzzy controller can be achieved. The derivation of the fuzzy controller model is performed by employing the states of the process fuzzy model. However, the estimation of the parameters of the fuzzy controller are achieved by minimising the error between the reference power P_r and the controlled power $P_g(t)$, thus leading to maximise the wind turbine generated power.

With reference to stable fuzzy systems, whose inverted dynamics are also stable, a nonlinear controller can be simply designed by inverting the fuzzy model itself. Moreover, when modelling errors and disturbances are not present, this controller is able to allow for exact tracking with zero steady–state errors. However, modelling errors and disturbance effects are always present in real conditions, which can be tackled by directly identifying the controller model (*i.e.* the inverse controlled model) using the FMID approach. Differently from [22] , a robust control strategy is thus achieved by minimising a cost function which includes the difference between the desired and controller outputs, and a penalty on the system stability. In general, a nonconvex optimisation problem has to be solved, which hampers the direct application of the proposed approach. However, the optimisation scheme described in [24] can be exploited, which is based on a parametrised search technique applied at a higher level to formulate the control objectives and constraints.

Note that, as remarked in Section 2, the fuzzy approach proposed in this work is able to provide a high–fidelity description of the wind turbine behaviour, which already includes uncertainty and disturbance, as described *e.g.* in [17]. The fuzzy approach is thus used

again to derive the formulation of the controller in the form of TS prototypes. The parameters of the controller fuzzy model are estimated by minimising the difference between the monitored outputs and the reference ones, taking into account the disturbance and the uncertainty affecting the wind turbine process. Therefore, the approach proposed in this work is able to cope with external disturbance modelled in the wind turbine benchmark.

In this way, the estimated controller based on the process inverse model and approximated via a fuzzy prototype is able to describe the complete behaviour of the monitored plant in its different working conditions (*i.e.* partial and full load situations). In fact, the rule–based fuzzy inference system of Eq. (14) has been derived for modelling the wind turbine dynamic process of Eq. (9) in its equivalent discrete–time form of Eq. (15):

$$y(t_{k+1}) = f\left(\boldsymbol{x}(t_k), u(t_k)\right) \tag{15}$$

and, in particular, the TS fuzzy representation has the form of Eq. (16) :

$$y(t_{k+1}) = \frac{\sum_{i=1}^{M} \mu_i^{(m)}\left(\boldsymbol{x}^{(m)}(t_k)\right)\left(\boldsymbol{a}_i^{(m)}\boldsymbol{x}^{(m)}(t_k) + b_i^{(m)}\right)}{\sum_{i=1}^{M} \mu_i^{(m)}\left(\boldsymbol{x}^{(m)}(t_k)\right)} \tag{16}$$

The current state $\boldsymbol{x}(t_k) = [y(t_k),\ldots,y(t_{k-n+1}),u(t_k),\ldots,u(t_{k-n+1})]^T$ and the input $u(t_{k+1})$ represent the inputs that drive the model of Eq. (16) . Its output represents the prediction of the system output at the next sample $y(t_{k+1})$. The model of Eq. (16) requires the estimated membership functions $\mu_i^{(m)}$, the state $\boldsymbol{x}^{(m)}$ and the parameters $\boldsymbol{a}_i^{(m)}$, $b_i^{(m)}$ of the controlled system, which are denoted by the superscript (m).

Therefore, the input $u(t_k)$ generated by the control law feeds the monitored process such that its output $y(t_{k+1})$ asymptotically follows the desired (reference) output $r(t_{k+1})$. This behaviour is obtained by using the inverse model principle, represented by the expression of Eq. (17):

$$u(t_{k+1}) = f^{-1}\left(\boldsymbol{x}^{c}(t_k), r(t_{k+1})\right) \tag{17}$$

that is a nonlinear function of the vector $\boldsymbol{x}^{c}(t_k)$ and the reference $r(t_{k+1})$.

However, in general, with reference to Eq. (17) , it is difficult to determine the analytical expression of the inverse function $f^{-1}(\cdot)$. Therefore, the methodology proposed in this work suggested to exploit the identified fuzzy TS prototype of Eq. (16) to provide the particular state $\boldsymbol{x}^{(m)}(t_k)$ at each time sample t_k. In this way, from this relation, the inverse mapping $u(t_{k+1}) = f^{-1}\left(\boldsymbol{x}^{(c)}(t_k), r(t_{k+1})\right)$ is directly identified the form of Eq. (14) , if the controlled system is stable, and in particular in the form of Eq. (18) :

$$u(t_{k+1}) = \frac{\sum_{i=1}^{M} \mu_i^{(c)}\left(\boldsymbol{x}^{(c)}(t_k)\right)\left(\boldsymbol{a}_i^{(c)}\boldsymbol{x}^{(c)}(t_k) + b_i^{(c)}\right)}{\sum_{i=1}^{M} \mu_i^{(c)}\left(\boldsymbol{x}^{(c)}(t_k)\right)} \tag{18}$$

where the state $\boldsymbol{x}^{(c)}(t_k) = [\boldsymbol{x}^{(m)}(t_k), r(t_k),\ldots,r(t_{k-n+1})]^T$ and the reference signal $r(t_{k+1})$ represent the inputs of the identified controller model. The model of Eq. (18) contains the estimated membership functions $\mu_i^{(c)}$ and the parameters $\boldsymbol{a}_i^{(c)}$, $b_i^{(c)}$ of the identified controller model, that are denoted by the superscript (c). The complete scheme is outlined in Figure **5**.

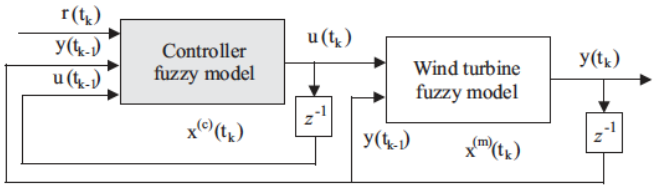

Figure 5: The fuzzy controller based on the inverse process model principle.

Note that Figure **4** sketches the general principle of the design of the controller for a system with input $u(t_k)$ and output $y(t_k)$. On the other hand, the signal $r(t_k)$ represents the generic set–point to be tracked (*i.e.* P_r or ω_{nom}) depending on the working region (1 or 2) and the controlled output (*i.e.* the sampled signal $P_g(t)$ or $\omega_g(t)$). Under this assumption, the identification of the fuzzy controller parameters leads to minimise the difference between $r(t_k)$ and $y(t_k)$. It is not required that $r(t_k)$ equals $y(t_k)$, thus making the problem feasible within the selected degree of accuracy.

Figure **5** highlights the series connection between the controller models (*i.e.* the inverse process model identified using the fuzzy systems) and the process model itself (described by means of fuzzy models), which should lead to an identity mapping as in Eq. (19):

$$
\begin{aligned}
&y(t_{k+1}) = f\left(\boldsymbol{x}^{(m)}(t_k), u(t_{k+1})\right) = \\
&= f\left(\boldsymbol{x}^{(m)}(t_k), f^{-1}\left(\boldsymbol{x}^{(c)}(t_k), r(t_{k+1})\right)\right) = \\
&= r(t_{k+1})
\end{aligned} \tag{19}
$$

where $r(t_{k+1}) = f\left(\boldsymbol{x}^{(m)}(t_k), u(t_{k+1})\right)$ for a proper value of $u(t_k)$. However, the expression of Eq. (19) holds in ideal conditions. Moreover, the model–reality mismatch and measurement errors are properly managed by means of the fuzzy modelling scheme recalled in Section 3. In this way, the difference $|r(t_{k+1}) - f\left(\boldsymbol{x}^{(m)}(t_k), u(t_{k+1})\right)|$ can be made arbitrarily small by a suitable selection of the model parameters, *i.e.* the fuzzy membership functions $\mu_i^{(c)}$, the number of clusters M, and the regressand $\boldsymbol{a}_i^{(c)}$, $b_i^{(c)}$.

Moreover, as highlighted in Figure **5**, the fuzzy model of the process is used for providing the state vector $\boldsymbol{x}^{(m)}(t_k)$. Therefore, the state of the fuzzy controller $\boldsymbol{x}^{(c)}(t_k)$ is updated using the process model state $\boldsymbol{x}^{(m)}(t_k)$ and the reference input $r(t_k)$. These computations are performed using standard matrix operations, thus making the algorithm suitable for real–time implementations [25] .

As already remarked, the effects of the model uncertainty and disturbance lead to a different behaviour of the model with respect to controlled process, thus resulting in a mismatch between the process outputs $y(t_k)$ and their references $r(t_k)$. This mismatch can be compensated by means of the on–line mechanism described by the expressions of Eqs. (16) and (18) . These issues motivate the model–based strategy relying on the adaptive algorithm proposed in Section 3.2.

Note finally that the fuzzy controller proposed in this section and depicted in Figure **5** will replace the baseline wind turbine regulator of Section 2 and reported in Figure **4**.

3.2. Adaptive Control Scheme

This section describes the model–based adaptive control strategy used in connection with the on–line estimation scheme presented above. In more detail, with reference to the wind turbine system recalled in Section 2, adaptive controllers for second order models are designed. Moreover, the considered adaptive controllers are based on the trapezoidal method of discretisation.

An on–line version of the batch Frisch scheme estimation methodology is recalled here estimating the parameters of dynamic EIV models. For the derivation of the adaptation law, an on–line bias–compensating algorithm is also implemented. Thus, the on–line Frisch scheme estimation is generalised to enhance its applicability to real–time implementations. Moreover, by means of an exponential forgetting factor included in the adaptation law, the algorithm is able to deal with Linear Parameter–Varying (LPV) structures, that are exploited in connection with the model–based design of the adaptive control scheme, presented in Section 3.2. Note that the adaptive algorithm proposed here exploits an iterative procedure that starts from an initial controller estimated off–line, for example using the baseline controller already implemented in the wind turbine simulator, and described in Section 2. This initial controller model is subsequently adapted on–line using the recursive laws in order to track the different operating conditions of the process under investigation.

Thus, the considered scheme is proposed for the on–line identification of the process modelled by the following transfer function $G(z)$:

$$
G(z) = \frac{A(z^{-1})}{B(z^{-1})} = \frac{b_1 z^{-1} + \ldots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + \ldots + a_{n_a} z^{-n_a}} \tag{20}
$$

where a_i, b_i, n_a, and n_b represent the unknown parameters and the structure of the model, defining the polynomials $A(z^{-1})$ and $B(z^{-1})$, whilst z is the discrete–time complex variable.

The parameter vector describing the linear relationship is given by:

$$
\theta = \left[a_1 \ldots a_{n_a} \, b_1 \ldots b_{n_b} \right]^T \tag{21}
$$

whose extended version is defined as in Eq. (22) :

$$
\overline{\theta} = \left[1 \theta^T \right]^T \tag{22}
$$

An equivalent expression of the considered relations is obtained by using vector and matrix notations, in the form of Eq. (23) :

$$
\psi^T(t_k)\overline{\theta} = 0 \tag{23}
$$

where the regressor vector $\psi(t_k)$ is defined as:

$$
\begin{aligned}
\psi(t_k) = \; & [-y(t_k), -y(t_{k-1}), \ldots, -y(t_{k-n_a}), \ldots \\
& u(t_{k-1}), \ldots, u(t_{k-n_b})]^T
\end{aligned} \tag{24}
$$

with t_k the sample time, $t_k = kT_s$, and $k = 0,1,\ldots,N$.

The Frisch scheme provides the estimates of the measurement errors affecting the input and output signals $u(t_k)$ and $y(t_k)$, *i.e.* σ_u and σ_y, and θ for a linear time–invariant dynamic system. Note that the polynomial orders n_a and n_b in the relation of Eq. (20) are assumed to be fixed in advance.

From the Frisch scheme method, the following expression is considered:

$$\left(\Sigma_\psi - \Sigma_{\tilde{\psi}}\right)\overline{\theta} = 0 \tag{25}$$

where the noise covariance matrix is given by:

$$\Sigma_{\tilde{\psi}} = \begin{bmatrix} \sigma_y I_{n_a+1} & 0 \\ 0 & \sigma_u I_{n_b} \end{bmatrix} \tag{26}$$

which are approximated by the sample covariance matrix over N samples:

$$\Sigma_{\tilde{\psi}} \approx \frac{1}{N}\sum_{k=1}^{N}\psi(t_k)\psi^T(t_k) \tag{27}$$

Thus, the Frisch scheme aims at providing suitable noise variances σ_u and σ_y such that $\left(\Sigma_\psi - \Sigma_{\tilde{\psi}}\right)$ results to be a matrix singular positive semidefinite as it is rank–one deficient. On the other hand, the system represented by the expression of Eq. (25) can be solved, and $\overline{\theta}$ represents its solution.

The expression of Eq. (28) is determined:

$$\varepsilon(t_k)\left(\overline{\theta}\right) = A(z^{-1})y(t_k) - B(z^{-1})u(t_k) \tag{28}$$

whilst the so–called sample auto–covariance is defined in the form of Eq. (29) :

$$r_{\varepsilon t_h, t_N} = \frac{1}{N}\sum_{l=1}^{N}\varepsilon_l\left(\overline{\theta}\right)\varepsilon(t_{l+h})\left(\overline{\theta}\right) \tag{29}$$

where the subscripts l and h in Eq. (29) indicate time–shifts.

The on–line control development requires a recursive estimate of the model parameters represented by the vector $\theta(t_k)$ of Eq. (20) , while the input and output data $u(t_k)$ and $y(t_k)$ acquired on–line by the dynamic process of the wind turbine system. In fact, the adaptive control law computed at time step k is based on the recursive estimate of a model of the process, which is derived exploiting the dynamic data up to the sample k. In this way, the algorithm of the Frisch scheme defined by the expressions of Eqs. (25), (27), and (29) is expressed by means of an on–line scheme.

Note that the expressions of Eqs. (27) and (29) are required in their recursive form. Therefore, whilst the derivation of the on–line form of the covariance matrix update is easily obtained as in the form of Eq. (30):

$$\Sigma_{\tilde{\psi}}(t_k) = \frac{k-1}{k}\Sigma_{\tilde{\psi}}(t_k) + \frac{1}{k}\psi(t_k)\psi^T(t_k) \tag{30}$$

the formulation of the auto–covariance expression $r_{\varepsilon t_h, t_k}$ can be obtained recursively for $1 \le l \le k$ only if the approximated expression of Eq. (31) is considered:

$$\varepsilon(t_l)\left(\overline{\theta}(t_k)\right) \approx \varepsilon(t_l)\left(\overline{\theta}(t_l)\right) \tag{31}$$

for $l < k$. In this way, only the residual $\varepsilon(t_k)\left(\overline{\theta}(t_k)\right)$ has to be computed at t_k using the lagged data in the vector $\psi(t_k)$ and the updated estimate $\overline{\theta}(t_k)$ of the model parameters. The on–line computation of the expression of the auto–covariance matrix of Eq. (32):

$$\begin{aligned} r_{\varepsilon t_h, t_k} &= \frac{k-1}{k} r_{\varepsilon t_k, t_{k-1}} + \\ &+ \frac{1}{k}\varepsilon(t_k)\left(\overline{\theta}(t_k)\right)\varepsilon(t_{k+h})\left(\overline{\theta}(t_k)\right) \end{aligned} \tag{32}$$

can be achieved using only the vector $\varepsilon(t_{k+h})\left(\overline{\theta}(t_k)\right)$ at each samples. The initial values θ_0, $\Sigma_{\tilde{\psi}0}$, and $r_{\varepsilon 0,h}$ for the recursive algorithm are equal to the variables of the classic Frisch scheme batch procedure.

Since variations of system properties have to be tracked on–line, in order to cope with time–varying systems, this paper considers a further modification of the recursive estimation scheme. This point can be achieved by placing more emphasis on the more recent data, while forgetting the older ones. Therefore, the methodology represented by the expressions of Eqs. (30) and (32) with the approximation of Eq. (31) is implemented by including the so–called exponential forgetting factor. This is achieved in practice by defining the new expressions of the sample covariance and auto–covariance matrices in the form of Eqs. (33) :

$$\begin{cases} H_{\Sigma_{\tilde{\psi}}}(t_k) &= \omega(\delta)\Sigma_{\tilde{\psi}}(t_k) \\ h_{\varepsilon t_h, t_k} &= \omega(\delta) r_{\varepsilon t_h, t_k} \end{cases} \tag{33}$$

where $\omega(\delta)$ is a scaling factor. In this way, the updated expressions have the form:

$$\begin{cases} H_{\Sigma_{\tilde{\psi}}}(t_k) &= (1-\delta)H_{\Sigma_{\tilde{\psi}}}(t_{k-1}) + \\ & \quad +\delta\psi(t_k)\psi^T(t_k) \\ h_{\varepsilon t_h,t_k} &= (1-\delta)h_{\varepsilon t_h,t_{k-1}} + \\ & \quad +\delta\varepsilon(t_k)\left(\overline{\theta}(t_k)\right)\varepsilon(t_{k+h})\left(\overline{\theta}(t_k)\right) \end{cases} \quad (34)$$

with $0<\delta<1$ representing the forgetting factor. Thus, the adaptive Frisch scheme algorithm is implemented via Eqs. (34) in three steps. First, θ_0, $\Sigma_{\tilde{\psi}0}$, and $r_{\varepsilon t_0,t_h}$ with $h \le n_a$ are initialised. Moreover, at each recursion step, by means of $r_{\varepsilon t_h,t_k}$, the noise variances σ_u and σ_y are computed. Finally, at each recursion step, $\overline{\theta}(t_k)$ is determined by solving Eq. (25) via the expression of Eq. (34) . In this way, the vector $\theta(t_k)$ contains the estimates of the model parameter derived at time t_k.

Once the parameters $\theta(t_k)$ of the discrete–time linear time–varying model of the nonlinear dynamic process of Eq. (9) have been computed at each time t_k, the adaptive controller is derived as follows.

With reference to Eq. (20) , the transfer function of the time–varying controlled system with $n_a = n_b = n = 2$ is considered, whose parameters estimated using the on–line identification approach recalled above:

$$\theta(t_k) = \left[\hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2\right]^T \quad (35)$$

The control law corresponding to the discrete–time adaptive controller in its difference form of Eq. (36) :

$$\begin{cases} \Delta e(t_k) &= e(t_k) - e(t_{k-1}) \\ u(t_k) &= K_p\left[\Delta e(t_k) + \dfrac{T_s}{T_I}\dfrac{\Delta e(t_k)}{2}\right] + u(t_{k-1}) \end{cases} \quad (36)$$

with $e(t_k)$ representing the tracking error, $e(t_k) = r(t_k) - y(t_k)$, and $r(t_k)$ the reference (set–point) signal.

The controller parameters K_p and T_I are here time–varying and derived from the on–line model parameters in the vector $\theta(t_k)$. The control law can be represented also in its feedback formulation as described by Eq. (37) :

$$u(t_k) = q_0 e(t_k) + q_1 e(t_{k-1}) + u(t_{k-1}) \quad (37)$$

where the new controller variables q_0 and q_1 (or K_p and T_I) are derived from the relations of Eq. (38) :

$$\begin{cases} q_0 &= K_p\left(1 + \dfrac{T_s}{2T_I}\right) \\ q_1 &= -K_p\left(1 - \dfrac{T_s}{2T_I}\right) \end{cases} \quad (38)$$

where the parameters K_p and T_I are functions of the (time–varying) critical gain and the critical period of oscillations, respectively, K_{P_u} and T_u :

$$K_p = 0.6K_{P_u}, T_I = 0.5T_u \quad (39)$$

that depend on the time–varying model parameters in the vector $\theta(t_k)$. In particular, when considering a second order model described by its (time–varying) parameters $\hat{a}_2$, $\hat{a}_1$, $\hat{b}_2$, and $\hat{b}_1$, the variables K_{P_u} and T_u required by the Ziegler–Nichols method used in this work are computed at each time step k from the following relations [14, 2] :

$$\begin{cases} K_{P_u} = \dfrac{\hat{a}_1 - \hat{a}_2 - 1}{\hat{b}_2 - \hat{b}_1} \\ T_u = \dfrac{2\pi T_s}{arccos\,\gamma}, \qquad with\ \gamma = \dfrac{\hat{a}_2\hat{b}_1 - \hat{a}_1\hat{b}_2}{2\hat{b}_2} \end{cases} \quad (40)$$

In this way, the adaptive discrete–time linear controllers of Eq. (36) or (37) are designed on the basis of the time–varying linear model of Eq. (20) estimated via the on–line identification scheme from the data of the nonlinear wind turbine process of Eq. (9) .

The adaptive regulators considered in this section have been implemented in the Simulink environment, integrating also the on–line estimation scheme recalled above.

The experimental set–up employs 3 adaptive regulators used for the control of the blade pitch angles, and the generator control torque, in the partial and full load working conditions. The complete block scheme is shown in Figure **6**.

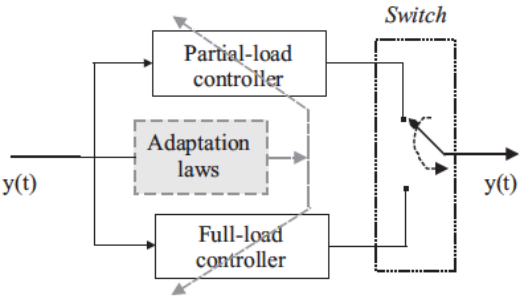

Figure 6: Layout of the adaptive control strategy.

Moreover, the adaptive control scheme represented in Figure **6** will replace the baseline wind turbine controller recalled in Section 2 and depicted in Figure **3**. In this way, the adaptive controller should be able to manage possible uncertainty affecting the wind turbine system, thus allowing to improve the performance of the baseline wind turbine control described in Section 2.

Note that a bumpless transfer mechanism has been considered along with the implementation of the switching logic among the partial and full load controllers, as shown in Figure 5. However, this mechanism is not included in the schemes shown in this paper, but it was developed as described *e.g.* in [6, 5] .

Finally, Section 4 will show the achieved results regarding the design and the application of the adaptive controller to the data from the wind turbine benchmark.

4. SIMULATIONS, PERFORMANCE ANALYSIS, AND COMPARISONS

This section presents the simulation results achieved with the proposed control techniques relying on both the data–driven and model–based fuzzy modelling technique, which are oriented to the identification of the fuzzy controller description and the adaptive control strategy using the on–line estimated models. The simulations obtained with these regulators are summarised in Section 4.1. Moreover, the reliability and robustness analysis, followed by extended comparisons with respect to different control solutions are reported in Sections 4.2 and 4.3, respectively.

4.1. Controller Performance Tests

Regarding the fuzzy modelling and identification method, the GK clustering algorithm recalled in Section 3 with a number $M = 3$ of clusters and delays $n = 2$. These variables were applied for clustering the first data set consisting of $\left\{P_g(t_k), \omega_g(t_k), \beta_r(t_k)\right\}$. A number of samples $k = 1,2,\ldots,N$ were considered with $N = 440 \times 10^3$. The same number of clusters and shifts were exploited for clustering the second data set $\left\{P_g(t_k), \omega_g(t_k), \tau_g(t_k)\right\}$. After this procedure, the structures of the TS prototypes were derived for each output $y(t_k)$ equal to $P_g(t_k)$ and $\omega_g(t_k)$. In this way, the 2 continuous–time outputs $y(t) = \left[\omega_g(t), \tau_g(t)\right]$ of the wind turbine continuous–time model of Eq. (9) are approximated by 2 TS fuzzy prototypes of Eq. (14) .

The performances of the fuzzy models that are derived using the procedure described above can be evaluated using the so–called Variance Accounted For (VAF) parameter [1]. In particular, the TS fuzzy model reconstructing the first output has a VAF index bigger than 90%, whilst for the second one it was higher than 99%. This means that the fuzzy prototypes are able to describe the behaviour of the controlled process with very good precision. These estimated TS fuzzy models have been used for the derivation of the fuzzy controllers and applied to the considered wind turbine benchmark.

Two fuzzy controllers with 2 inputs and 1 output have been used for the control of the wind turbine system. As shown in Figure **4**, these controllers are both fed by the sampled signals $P_g(t_k)$, $\omega_g(t_k)$ (*i.e.* the outputs of the wind turbine system) for the generation of the sampled signals $\beta(t_k)$ and $\tau_r(t)$ (*i.e.* the control inputs for the wind turbine system). By using the inverse model principle, they were estimated exploiting the methodology recalled in Section 3.1. Again, the GK fuzzy clustering method has lead to 2 fuzzy regulators applied to the data sets $\left\{\beta(t_k), P_g(t_k), \omega_g(t_k)\right\}$ and $\left\{\tau_g(t_k), P_g(t_k), \omega_g(t_k)\right\}$, respectively, with $M = 3$ clusters and $n = 3$ lagged signals.

The controller performances were verified and validated via extensive simulations by considering different data sequences generated via the wind turbine simulator. Table **2** reports the values of the per–cent Normalised Sum of Squared tracking Error ($NSSE\%$) index defined in Eq. (41):

$$NSSE\% = 100 \sqrt{\frac{\sum_{k=1}^{N} \left(r(t_k) - y(t_k)\right)^2}{\sum_{k=1}^{N} r^2(t_k)}} \tag{41}$$

Note that in partial load operation (region 1, $\beta_r = 0$), the performance is represented by the comparison between the power produced by the generator, $y(t_k) = P_g(t_k)$, with respect to the theoretical maximum power output, $r(t_k) = P_r$. On the other hand, in full load operation (region 2), the tracking error is given by the difference between the generator speed, $y(t_k) = \omega_g(t_k)$ and its nominal value, $r(t_k) = \omega_{nom}$. The achieved results are shown in Figure **7** for the case of the identified fuzzy controllers.

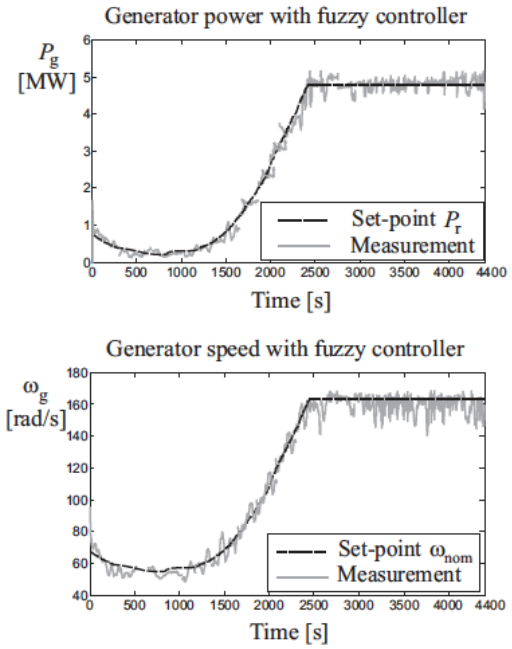

Figure 7: Generator speed ω_g and power P_g (bold gray line) with respect to their references (dashed black line) ω_{nom} and P_r with the fuzzy controllers.

Figure **7** depicts the signal representing the controlled generator speed ω_g and the generated power P_g in gray bold gray line with respect to their desired values ω_{nom} and P_r in dashed black line, respectively. It can be noted that in both partial and full load conditions, the fuzzy controllers are able to track the reference signals, as recalled in Section 2. Note that the performance of the fuzzy regulators are better than those achieved via the baseline governors, which were tuned with frequency approaches described in [6, 5] .

With reference to the second adaptive design approach using adaptive solutions, the two outputs $P_g(t)$ and $\omega_g(t)$ of the wind turbine continuous–time nonlinear model of Eq. (9) were approximated by 2 second–order time–varying discrete–time models of Eq. (20) with 2 inputs and 1 output. Using these 2 LPV prototypes, the model–based approach for determining the adaptive controllers recalled in Section 3.2 was exploited and applied to the wind turbine benchmark of Section 2. Thus, according to Section 3.2, the parameters of the adaptive controllers were computed on–line. In particular, for each output, 2 second–order ($n_a = n_b = 2$) time–varying prototypes were identified, and the adaptive regulator parameters in Eqs. (36) or (37) were computed analytically at each time step k. Also in this case, with reference to the adaptive controller structure of Eqs. (36) or (37) , the parameters of the adaptive controllers were tuned on–line via the Ziegler–Nichols rules, applied to the LPV models. This adaptive procedure is already implemented and available in [14, 2] . In this way, if both the model on–line parametric identification and the regulator recursive tuning procedure are exploited, the parameter adaptation mechanisms should lead to good control performances.

The simulations with the adaptive regulators have been obtained in the same situation of the fuzzy controllers. In this case, 3 on–line regulators were exploited for the compensation of both the blade pitch angle $\beta(t)$ and the generator torque $\tau_g(t)$, in region 1 and region 2. The adaptive algorithm described above run with initial values for its parameters reported in Table **1**.

Table 2: Initialisation Parameters of the Adaptive Algorithm

Recursive Algorithm Parameter	Parameter Value
$\bar{\theta}(0)$	$[0.1, 0.15, 0.20, 0.250.30, 0.35]^T$
$\Sigma_{\bar{\psi}}(0)$	$10^{-1} I_7$
δ	0.995

Also with reference to the model–based adaptive approach, Figure 7 depicts both the controlled outputs P_g and ω_g in bold gray lines with respect to their reference values P_r and ω_{nom}, respectively, in dashed black lines. As it will be seen in the following, also for the case of the adaptive regulators, Figure **8** highlights that this approach leads to interesting performances.

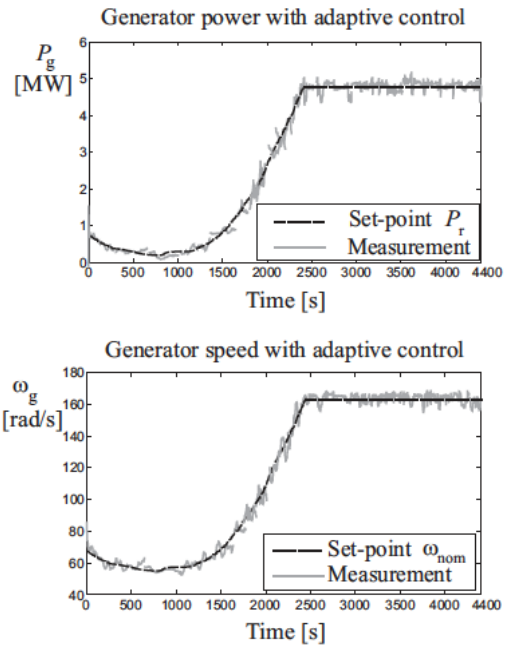

Figure 8: P_g and ω_g tracking capabilities in partial and full load conditions with the adaptive controllers.

Note that the recursive scheme is able to react actively with respect to variations of the working conditions of the wind turbine system. In fact, the fuzzy method can be considered a *passive* (even if robust) control solution, since the controllers are identified to passively tolerate the disturbance acting on the system. On the other hand, the adaptive methodology is able to counteract any variation or disturbance effects of the controlled process, thus representing an *active* control solution. These definitions were provided for different frameworks *e.g.* in [3, 30] .

In order to analyse the performance of the proposed adaptive strategy, Table **2** reports also the $NSSE$ values computed for these controllers.

Table 3: Controllers in Partial and Load Operations: $NSSE\%$ values

Controller Type	Partial Load	Full Load
Baseline governors	46.68%	20.96%
Fuzzy controllers	37.17%	17.85%
Adaptive controllers	28.73%	13.67%

According to the simulation results summarised in Table **2**, good tracking capabilities of the suggested adaptive controllers seem to be reached, and they are better than both the fuzzy regulators and the baseline governor, recalled in Section 2.

4.2. Robustness Analysis

This section summarises further simulation results that concern the evaluation of the achieved characteristics of the developed control strategies when the effects of uncertainty and disturbance are taken into account.

In particular, the wind turbine benchmark in the Matlab and Simulink environments can vary the variables and the parameters of the simulated process in a statistical way. In this way, it is possible to analyse the effects of the model–reality mismatch and the measurement errors on the designed controllers. Moreover, a Monte–Carlo analysis is also considered since it represents a practical approach for validating and verifying the features of the developed control schemes when applied to the considered wind turbine process. The same approach was for suggested for the first time by the same authors in [19] and applied to a different simulated system. The Monte–Carlo tool is very useful in this case since the behaviour of control strategies designed assuming the nominal plant depends on both the model–reality mismatch and the measurement errors.

Under these considerations, the uncertainty values of the parameters and variables of the wind turbine simulator considered in this work are summarised in Table **3**. Therefore, the Monte–Carlo analysis was achieved by modelling these parameters and variables as Gaussian stochastic processes, with mean values equal to the nominal ones, and standard deviations corresponding to realistic error values, typical of wind turbine models [16] .

Table 4: Wind Turbine Uncertain Variables

Model Variable Parameter	Standard Deviation
$\beta(t)$	11%
$\omega_g(t)$	18%
$\tau_g(t)$	21%
$P_g(t)$	20%
Pitch model parameters	49%
Drivetrain model efficiency	5%
Converter model time constant	50%

Therefore, for the evaluation of the reliability and robustness characteristics of the designed control schemes, the average values of the $NSSE\%$ index were computed and evaluated in simulation via 1000 Monte–Carlo runs. Note that, as already remarked in Section 3, proper algorithms were exploited for guaranteeing the derivation of controller models that lead to stable closed loops. On the other hand, the stability of the closed loop system when adaptive algorithms are exploited was investigated in [9] .

Note however that, if unstable models should be obtained due to large uncertainties of Table **3**, gain and phase margin requirements have to be included in the controller design, as described *e.g.* in [8] , and the controller parameters can be computed using the w–plane design. In this case, it can be shown that the model of Eq. (20) with $n_a = n_b = n = 2$ is transformed into its equivalent description of Eq. (42) :

$$\frac{T_s(b_1T_s - b_2T_s)w^2}{(a_1T_s^2 - T_s^2 - a_2T_s^2)w^2 + (4a_2T_s - 4T_s)w - 4 - 4a_2 - 4a_1} + \\ + \frac{\left(-2b_1T_s + 2b_2T_s + T_s(2b_1 + 2b_2)\right)w - 4b_1 - 4b_2}{(a_1T_s^2 - T_s^2 - a_2T_s^2)w^2 + (4a_2T_s - 4T_s)w - 4 - 4a_2 - 4a_1} \approx \\ \approx \frac{k}{w\tau_o - 1}e^{-wL} \tag{42}$$

Under the validity of the approximation of Eq. (42) , which neglects the fast dynamic stable modes, and consider only the unstable pole τ_o with the effective delay L (see *e.g.* the approximation in [26]), the adaptive controller parameters are computed via the relations of Eq. (43):

$$\begin{cases} K_p &= \dfrac{\delta \tau_o}{A_m k} \\ T_I &= \dfrac{1}{\dfrac{\pi}{2}\delta - \delta^2 L - \dfrac{1}{\tau_o}} \\ \delta &= \dfrac{A_m \varphi_m + \dfrac{\pi}{2} A_m (A_m - 1)}{(A_m^2 - 1)L} \end{cases} \tag{43}$$

The relations of Eq. (43) provide the parameters K_p and T_I of the adaptive regulator that allow to obtain the gain and phase margins (A_m, φ_m) for the identified (unstable) model of Eq. (42). Note that, even if approximations are used to derive the tuning formulas of (43), it can be seen that the achievable gain and phase margins can be quite close to the specified ones, and in general within a 5% of maximal error.

After these considerations, Table **4** reports the average $NSSE\%$ values by considering the effects on the input and output measurements given by the alteration of the model variables and parameters reported in Table **3**. Moreover, Table **4** shows how the considered control strategies, and especially the adaptive approach, is able to achieve excellent performances even in the presence of considerable error and uncertainty effects.

Table 5: Monte–Carlo Analysis for the Considered Control Schemes

Controller Strategy	Partial load $NSSE\%$	Full load $NSSE\%$
Baseline governors	48.23%	21.75%
Fuzzy controllers	37.19%	17.94%
Adaptive controllers	24.52%	13.72%

The achieved results highlight also that Monte–Carlo tool represents an effective and practical instrument for validating and verifying in simulation the design reliability and robustness of the considered control methodologies with respect to modelling uncertainty and measurement errors.

4.3. Performance Verification and Comparisons

The evaluation of the performances of the data–driven and model–based control strategies considered in this paper has been evaluated also on the basis of the following performance metrics, borrowed and modified from the fault diagnosis framework [19]:

• **False Tracking Rate** (FTR): the ratio between the total number of wrongly reference tracking and the number of simulations;

• **Missed Tracking Rate** (MTR): the ratio between the total number of missed reference tracking and the number of simulations;

• **Correct Tracking Rate** (CTR): the ratio between the number of correct reference tracking and the number of simulations;

• **Mean Tracking Delay** (MTD): the delay time between the reference tracking and the reference timing.

With reference to the indices above, note that the CTR index is complementary to MTR, since they refer to the tracking capabilities in the presence of uncertainty and disturbance. In contrast, the FTR index describes the tracking performance achieved only by the control designs, without considering any errors or anomalities occurring in the system. On the other hand, the MTD index considers the average delay occurring during the tracking of the reference signals.

Also in this case a proper Monte Carlo analysis has been performed in order to compute these performance metrics and to test the robustness of the considered control schemes. A set of 1000 Monte Carlo runs has been performed, during which realistic wind turbine uncertainties have been considered as described in Table **3**. Moreover, in addition to the considered fuzzy and adaptive strategies, the performance metrics of other control schemes are analysed.

The first alternative approach considered here uses a Support Vector machine based on a Gaussian Kernel (GKSV) originally developed in [13] and it was exploited here for control purpose. The scheme defines a vector of features for each working condition of the wind turbine, which contains relevant signals obtained directly from measurements, filtered measurements or their combinations. These vectors are subsequently projected onto the kernel of the Support Vector Machine (SVM), which provides suitable control sequences for all of the defined working conditions.

The second scheme consists in an Estimation–Based (EB) solution shown in [31]. In particular, a bank

of observers is designed to estimate the control signals that have to feed the controlled process. These observers were designed on the basis of a system linear model.

The third method relying on Up–Down Counters (UDC) was addressed in [18]. These tools, are commonly used in the aerospace framework, and they provide a different approach to the decision logic usually applied to the control. Indeed, the design of the control signals involves discrete–time dynamics and is not simply a function of the plant working conditions.

The fourth approach refers to Combined Observer and Kalman (COK) filter methods [4]. It relies on an observer used as a control signal residual generator, when the wind speed is considered a disturbance. This observer was designed to decouple the disturbance and simultaneously achieve optimal reference tracking in a statistical sense.

Finally, the fifth method is a General Fault Model (GFM) scheme, which is a method of automatic design [27]. The design strategy consists of three main steps. In the first step, a large set of potential controllers is designed. In the second step, the most suitable control signals to be included in the final system are selected. The third step tests the selected set of control laws, on the basis of extended comparisons of the estimated probability distributions of the tracking errors, evaluated with and without uncertainty or disturbance effects.

The results of the comparative analysis are summarised in Tables **6** and **7**, tacking into account the uncertainty effects reported in Table **4**. The different control approaches are analysed and compared.

The results summarised in Tables **6** and **7** serve to highlight the efficacy of the considered control solutions also with respect to different schemes. In details, both the data–driven and model–based approaches seem to work better than other approaches, and they have a noteworthy performance level considering the mean delay time, which is significantly low. Also the FTR and the MTR indices are lower than those of other approaches. However, for both model–based and

Table 6: Comparison of GKSV, EB, UDC and COK Control Strategies

Working Condition	Index	GKSV	EB	UDC	COK
Partial Load	FTR	0.234	0.224	0.123	0.003
	MTR	0.343	0.333	0.232	0.029
	CTR	0.657	0.667	0.768	0.971
	MTD (s)	47.24	44.65	69.03	19.32
Full Load	FTR	0.001	0.001	0.001	0.001
	MTR	0.002	0.003	0.002	0.003
	CTR	0.978	0.977	0.987	0.977
	MTD (s)	0.03	0.03	0.04	0.32

Table 7: Comparison of GFM, Fuzzy, Adaptive and Baseline Control Strategies

Working Condition	Index	GFM	Fuzzy	Adaptive	Baseline
Partial Load	FTR	0.235	0.001	0.018	0.403
	MTR	0.532	0.003	0.001	0.596
	CTR	0.468	0.997	0.999	0.404
	MTD (s)	13.74	0.08	0.08	70.87
Full Load	FTR	0.001	0.001	0.001	0.003
	MTR	0.002	0.001	0.001	0.003
	CTR	0.982	0.999	0.999	0.865
	MTD (s)	0.05	0.02	0.01	0.89

data–driven designs, optimisation stages are required, for example for the selection of the GK clustering algorithm. Furthermore, the GKSV approach presents quite high delays, with big FTR and MTR. EB has comparable performance with respect to GKSV in terms of FTR, CTR and MTR, but with lower MTD. UDC can show quite high FTR in both the working conditions. COK and GFM have similar performances, with important MTD, FTR and MTR. However, in general, the proposed data–driven and model–based approaches are able to achieve good tracking capabilities, with minimum MTD, and higher CTR with respect to the other control methodologies.

5. CONCLUSION

The work addressed two control examples for a wind turbine dynamic simulator, since it was proposed as benchmark representing a complex dynamic system driven by stochastic disturbances and uncertain load conditions. Moreover, the aerodynamic models of these processes is nonlinear, thus making their modelling a challenging problem. Therefore, the design of control strategies for these complex processes has to consider these aspects. In this way, the paper analysed the design of two data–driven and model–based control methodologies, which represented viable, reliable, and robust control schemes for the proposed wind turbine benchmark. Experiments with the wind turbine simulator and the Monte–Carlo tool were the practical instruments for assessing the most important characteristics of the developed control methodologies, when the model–reality mismatch and measurement errors were also considered. The analysed control methods were finally compared with respect to different control solutions proposed in the related literature, in order to highlight advantages and drawbacks of the developed strategies. The obtained results showed that the considered solutions represent viable, robust and reliable control applications to real wind turbine systems.

FUNDING

This work was not financially supported by any grant.

REFERENCES

[1] R. Babuška, Fuzzy Modeling for Control. 1em plus 0.5em minus 0.4em Boston, USA: Kluwer Academic Publishers, 1998.

[2] V. Bobál and P. Chalupa, Self-Tuning Controllers Simulink Library, Tomas Bata University in Zlín, Faculty of Technology, Zlín, Czech Republic, 2002, http://www.utb.cz/stctool/.

[3] J. Chen and R. J. Patton, Robust Model-Based Fault Diagnosis for Dynamic Systems. 1em plus 0.5em minus 0.4em Boston, MA, USA: Kluwer Academic Publishers, 1999.

[4] W. Chen, S. X. Ding, A. H. A. Sari, A. Naik, A. Q. Khan, and Y. S., "Observer-based FDI schemes for wind turbine benchmark," in Proceedings of the 18th IFAC World Congress 2011, vol. 18, no. 1, Milan, Italy, 28 Aug. - 2 Sept. 2011, pp. 7073-7078. https://doi.org/10.3182/20110828-6-IT-1002.03469

[5] T. Esbensen, C. Sloth, M. Odgaard Niss, and B. Thorarins Jensen, "Fault Diagnosis and Fault Tolerant Control of Wind Turbines," Department of Electronic Systems, Aalborg University, Aalborg, Denmark, Technical Report 09gr1030, June 2009.

[6] T. Esbensen and C. Sloth, "Fault Diagnosis and Fault Tolerant Control of Wind Turbines," Department of Electronic Systems, Aalborg University, Aalborg, Denmark, Technical Report 08gr830, June 2008.

[7] V. Galdi, A. Piccolo, and P. Siano, "Designing an adaptive fuzzy controller for maximum wind energy extraction," IEEE Transactions on Energy Conversion, vol. 23, no. 2, June 2008. https://doi.org/10.1109/TEC.2007.914164

[8] W. K. Ho and W. Xu, "PID tuning for unstable processes based on gain and phase-margin specifications," IEE Proceedings - Control Theory and Applications, vol. 145, no. 5, pp. 392-396, September 1998. https://doi.org/10.1049/ip-cta:19982243

[9] K. E. Johnson, L. Y. Pao, M. J. Balas, V. Kulkarni, and L. J. Fingersh, "Stability analysis of an adaptive torque controller for variable speed wind turbines," in Proceedings of the 43rd IEEE Conference on Decision and Control - CDC'04, vol. 4, Paradise Island, Bahamas, December 14-17 2004, pp. 4087-4094. https://doi.org/10.1109/CDC.2004.1429392

[10] 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, DOI: 10.1109/MCS.2006.1636311. https://doi.org/10.1109/MCS.2006.1636311

[11] A. Juditsky, H. Hjalmarsson, A. Beneviste, L. Delyon, B. Ljung, J. Sjöberg, and Q. Zhang, "Nonlinear black-box modelling in system identification: a mathematical foundation," Automatica, vol. 31, no. 12, pp. 1691-1724, 1995. https://doi.org/10.1016/0005-1098(95)00120-8

[12] Y. Landau, Adaptive Control. 1em plus 0.5em minus 0.4em 270 Madison Avenue, New York: Marcel Dekker, 1979, iSBN 0-8247-6548-6.

[13] N. Laouti, N. Sheibat-Othman, and S. Othman, "Support vector machines for fault detection in wind turbines," in Proceedings of the 18th IFAC World Congress 2011, vol. 18, no. 1, Milan, Italy, 28 Aug. - 2 Sept. 2011, pp. 7067-7072. https://doi.org/10.3182/20110828-6-IT-1002.02560

[14] J. Machácek and V. Bobál, "Adaptive PID controller with on-line identification," Journal of Electrical Engineering, vol. 53, pp. 233-240, 2002.

[15] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault-Tolerant Control of Wind Turbines: A Benchmark Model," IEEE Transactions on Control Systems Technology, vol. 21, no. 4, pp. 1168-1182, July 2013, ISSN: 1063-6536. https://doi.org/10.1109/TCST.2013.2259235

[16] P. F. Odgaard and J. Stoustrup, "A benchmark evaluation of fault tolerant wind turbine control concepts," IEEE Transactions on Control Systems Technology, vol. 23, no. 3, pp. 1221-1228, April 2015. https://doi.org/10.1109/TCST.2014.2361291

[17] 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, no. 1, Barcelona, Spain, June 30 - July 3 2009, pp. 155-160. https://doi.org/10.3182/20090630-4-ES-2003.00026

[18] A. A. Ozdemir, P. Seiler, and G. J. Balas, "Wind turbine fault detection using counter-based residual threshold-ing," in Proceedings of the 18th IFAC World Congress 2011, vol. 18, no. 1, Milan, Italy, 28 Aug. - 2 Sept. 2011, pp. 8289-8294. https://doi.org/10.3182/20110828-6-IT-1002.01758

[19] R. J. Patton, F. J. Uppal, S. Simani, and B. Polle, "Robust FDI applied to thruster faults of a satellite system," Control Engineering Practice, vol. 18, no. 9, pp. 1093-1109, September 2010, ACA'07 - 17th IFAC Symposium on Automatic Control in Aerospace Special Issue. Publisher: Elsevier Science. ISSN: 0967-0661. https://doi.org/10.1016/j.conengprac.2009.04.011

[20] J. E. Slotine and W. Li, Applied Nonlinear Control. 1em plus 0.5em minus 0.4em Prentice-Hall, 1991.

[21] H. Schulte and S. Georg, "Nonlinear control of wind turbines with hydrostatic transmission based on takagi-sugeno model," Journal of Physics: Conference Series, vol. 524, no. 012085, pp. 1-8, 2014. https://doi.org/10.1088/1742-6596/524/1/012085

[22] S. Simani, "Application of a Data-Driven Fuzzy Control Design to a Wind Turbine Benchmark Model," Advances in Fuzzy Systems, vol. 2012, pp. 1-12, November 2nd 2012, invited paper for the special issue: Fuzzy Logic Applications in Control Theory and Systems Biology (FLACE) . ISSN: 1687-7101, e-ISSN: 1687-711X. https://doi.org/10.1155/2012/504368

[23] S. Simani and P. Castaldi, "Data-Driven and Adaptive Control Applications to a Wind Turbine Benchmark Model," Control Engineering Practice, vol. 21, no. 12, pp. 1678-1693, December 2013, special Issue Invited Paper. ISSN: 0967-0661. PII: S0967-0661(13)00155-X. https://doi.org/10.1016/j.conengprac.2013.08.009

[24] S. Simani, C. Fantuzzi, R. Rovatti, and S. Beghelli, "Parameter identification for piecewise linear fuzzy models in noisy environment," International Journal of Approximate Reasoning, vol. 1, no. 22, pp. 149-167, September 1999, Publisher: Elsevier. https://doi.org/10.1016/S0888-613X(99)00012-2

[25] R. Rovatti, C. Fantuzzi, and S. Simani, "High-speed DSP-based implementation of piecewise-affine and piecewise-quadratic fuzzy systems," Signal Processing Journal. Publisher: Elsevier, vol. 80, no. 6, pp. 951-963, June 2000, special Issue on Fuzzy Logic applied to Signal Processing. https://doi.org/10.1016/S0165-1684(00)00013-X

[26] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning," Journal of Process Control, vol. 13, no. 4, pp. 291-309, July 2003. https://doi.org/10.1016/S0959-1524(02)00062-8

[27] C. Svard and M. Nyberg., "Automated design of an FDI system for the wind turbine benchmark," in Proceedings of the 18th IFAC World Congress 2011, vol. 18, no. 1, Milan, Italy, 28 Aug. - 2 Sept. 2011, pp. 8307-8315. https://doi.org/10.3182/20110828-6-IT-1002.00618

[28] M. Zhao and J. C. Ji, "Nonlinear torsional vibrations of a wind turbine gearbox," Applied Mathematical Modelling, vol. 39, no. 16, pp. 4928-4950, August 2015. https://doi.org/10.1016/j.apm.2015.03.026

[29] "Dynamic analysis of wind turbine gearbox components," Energies, vol. 9, no. 110, pp. 1-18, February 2016, DOI: 10.3390/en9020110. https://doi.org/10.3390/en9020110

[30] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," Annual Reviews in Control, vol. 32, pp. 229-252, March 2008. https://doi.org/10.1016/j.arcontrol.2008.03.008

[31] X. Zhang, Q. Zhang, S. Zhao, R. M. G. Ferrari, M. M. Polycarpou, and T. Parisini, "Fault detection and isolation of the wind turbine benchmark: An estimation-based approach," in Proceedings of the 18th IFAC World Congress 2011, vol. 18, no. 1, Milan, Italy, 28 Aug. - 2 Sept. 2011, pp. 8295-8300. https://doi.org/10.3182/20110828-6-IT-1002.02808

[32] W. Zhao and K. Stol, "Individual Blade Pitch for Active Yaw Control of a Horizontal-Axis Wind Turbine," in Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit, AIAA. 1em plus 0.5em minus 0.4em Reno, NV, USA: AIAA, 8-11 January 2007. https://doi.org/10.2514/6.2007-1022

Received on 05-11-2022 Accepted on 29-11-2022 Published on 05-12-2022

DOI: https://doi.org/10.31875/2409-9694.2022.09.08