Chapter 11

Robust Control Applications to a Wind Turbine-Simulated System

Silvio Simani and Paolo Castaldi

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.71526

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 chapter is thus to provide two practical examples of development of robust control strategies when applied to a simulated wind turbine plant. Experiments with the wind turbine simulator represent the instruments for assessing the main aspects of the developed control methodologies.

Keywords: wind turbine simulator, data-driven and model-based approaches, fuzzy identification, online 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, while 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 [1].

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 task 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 chapter [2].

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 [1]. 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 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 speed. Blade pitch control is thus used also above the rated wind speed [1]. 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 [3].

Regarding the regulation strategies proposed in this chapter, two control design examples are described and applied to a wind turbine system. The wind turbine model exploited in this chapter 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 [2].

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. [4], fuzzy modelling and identification methods are able to directly provide nonlinear models from the measured input-output signals. Therefore, this chapter suggests to model the wind turbine plant via Takagi-Sugeno (TS) fuzzy prototypes [5], whose parameters are obtained by identification procedures. This approach is also motivated by previous works by the same authors [6].

Regarding the second *model-based* strategy presented in this chapter, it relies on an adaptive control scheme [7]. Again, with respect to pure nonlinear control methods [8], it does not require a detailed knowledge about the model structure. Therefore, this chapter 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 Simani and Castaldi [9], which makes use of exponential forgetting laws. This allows the online application of the Frisch scheme to derive the parameters of a time-varying controller.

Finally, the chapter 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.

2. Wind turbine simulator model

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.

The wind turbine system exploited in this chapter 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 Odgaard et al. [10]. A block scheme of the wind turbine simulator considered in this chapter is represented in **Figure 1**.

Both the generator speed and the generator power are controller by means of the two 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 signal $\omega_r(t)$ represents the rotor speed measurement, whereas $\omega_g(t)$ represents the converter velocity. Concerning the electric generator, $\tau_g(t)$ refers to its required torque, which is controlled by the converter. Therefore, this signal represents the measurement of the torque setpoint, $\tau_r(t)$. 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 Odgaard et al. [10].

The aerodynamic model reported in **Figure 1** is described as follows:

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

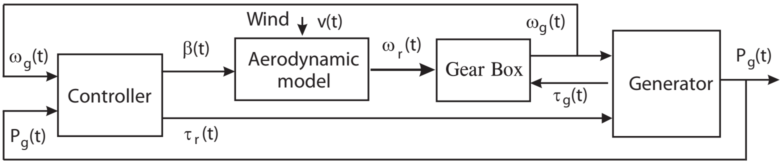

Figure 1. Scheme of the wind turbine process.

where the variable ρ represents the air density and A is the effective rotor area. Another important variable is represented by the so-called tip-speed ratio, which is defined as

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

with R the rotor radius. $C_p(\cdot)$ represents the power coefficient that is normally represented via a two-dimensional map [10]. The expression of Eq. (1) allows the computation of the signal $\tau_{aero}(t)$, by means of the estimated wind speed $v(t)$, and the measured $\beta(t)$ 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 (2) motivates the required reliable and robust control approaches suggested in this chapter.

A two-mass model is exploited to describe the drive-train system, while the hydraulic pitch system is modelled as second-order transfer function [10]. Moreover, the generator dynamics are described as a first-order transfer function. More details regarding the considered simulator are in Odgaard et al. [10]. Under these assumptions, the complete state-space description of the wind turbine model has the form of Eq. (3):

$$\begin{cases} \dot{x}_c(t) = f_c(x_c(t), u(t)) \\ y(t) = x_c(t) \end{cases} \tag{3}$$

where $u(t) = [\beta(t), \tau_g(t)]^T$ and $y(t) = x_c(t) = [P_g(t), \omega_g(t)]^T$ are the control inputs and the monitored output measurements, respectively, as shown in **Figure 1**. $P_g(t)$ is the generator power measurement, whereas $f_c(\cdot)$ represents the continuous-time nonlinear function that will be approximated via discrete-time models from N sampled data u_k and y_k, with the sample index $k = 1, 2, \ldots N$, as presented in Section 3. Finally, the model parameters and the map $C_p(\beta, \lambda)$ are chosen in order to represent a realistic turbine [10].

As described in Odgaard et al. [10], the baseline controller developed for this wind turbine system works in two normal operating conditions, namely, the region 1 corresponding to the power optimisation (partial load) and the region 2 of constant power production (full load). The partial load working condition (also known as working region 1), the optimal wind-power conversion is achieved without any pitching of the blades, which are fixed to 0°. 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 $\beta = 0$. Therefore, this working condition is completely defined by setting $\tau_g = \tau_r$ (i.e. the generator torque is equal to the reference one) with pitch angle $\beta = 0$.

The reference torque τ_r shown in **Figure 1** can be written as

$$\tau_r = K_{opt}\,\omega_r^2 \tag{4}$$

where:

$$K_{opt} = \frac{1}{2}\rho A R^3 \frac{C_{p_{\max}}}{\lambda_{opt}^3} \tag{5}$$

with $C_{p_{\max}}$ the maximal value of C_p, related the to λ_{opt}, i.e. the optimal tip-speed ratio.

When the power reference is achieved and the wind speed increases, the controller can be switched to the control region 2 (full load condition). In this zone, the control objective consists of tracking the power reference P_r, obtained by regulating β, 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; the second input of the controlled is τ_g.

The baseline controller considered in this chapter 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. 6 [10]:

$$\begin{cases} \beta_k = \beta_{k-1} + k_p e_k + \left(k_i T_s - k_p\right) e_{k-1} \\ e_k = \omega_{gk} - \omega_{nom} \end{cases} \tag{6}$$

with the sample index $k = 1, 2, \ldots, N$. 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 reported in [10].

To control the further input τ_g shown in **Figure 1**, a second PI regulator is used, in the form of Eq. (7):

$$\begin{cases} \tau_{rk} = \tau_{rk-1} + k_p e_k + \left(k_i T_s - k_p\right) e_{k-1} \\ e_k = P_{gk} - P_r \end{cases} \tag{7}$$

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

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. 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® simulator considered in this work includes also saturation blocks limiting the values of these control signals and their rates.

3. Data-driven and model-based designs

This section describes the two approaches considered in this chapter 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, whereas Section 3.2 proposes a different method relying on an adaptive technique.

The first method proposed in this chapter 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 [5]. This study proposes the use of TS fuzzy prototypes since they are able to model nonlinear dynamic systems with arbitrary accuracy [5]. 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 [5]. 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 Simani et al. [6]. 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. (8):

$$y_{k+1} = \frac{\sum_{i=1}^{M} \mu_i(x_k) y_i}{\sum_{i=1}^{M} \mu_i(x_k)} \tag{8}$$

where $y_i = a_i^T x + b_i$, with a_i the parameter vector (regressand) and b_i is the scalar offset. $x = x_k$ represents the regressor vector, which contains delayed samples of the signals u_k and y_k.

The antecedent fuzzy sets μ_i that determine the switching among the different submodels i are estimated from the data clusters [5]. The consequent parameters a_i and b_i are identified from the data by means of the methodology proposed in Simani et al. [6]. 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. (8) is used as predictor, since it derives the consequent parameters via the so-called Frisch scheme, developed for the errors-in-variables (EIV) structures [6].

Once the description of the monitored process is obtained in the form of Eq. (8), the data-driven approach for the design of the fuzzy controller proposed in this chapter is presented in Section 3.1.

The second approach exploited for obtaining the mathematical description of the wind turbine system under investigation is based on a recursive methodology, which will be used for the design of the second control strategy presented in Section 3.2. An online version of the batch Frisch scheme estimation methodology summarised above is recalled in the remainder of this section for estimating the parameters of dynamic EIV models. For the derivation of the adaptation law, an online bias-compensating algorithm is also implemented. Thus, the online 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.

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

$$G(z) = \frac{A(z^{-1})}{B(z^{-1})} = \frac{b_1 z^{-1} + \ldots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + \ldots + a_{n_a} z^{-n_a}} \tag{9}$$

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 $B(z^{-1})$, while z is the discrete-time complex variable.

The parameter vector describing the linear relationship is given by

$$\theta = \left[a_1 \ldots a_{n_a}\, b_1 \ldots b_{n_b}\right]^T \tag{10}$$

whose extended version is defined as in Eq. (11):

$$\overline{\theta} = \left[1\, \theta^T\right]^T \tag{11}$$

An equivalent expression of the considered relations is obtained by using vector and matrix notations, in the form of Eq. (12):

$$\psi_k^T \overline{\theta} = 0 \tag{12}$$

where the regressor vector ψ_k is defined as

$$\psi_k = \left[-y_k - y_{k-1} \ldots - y_{k-n_a}\, u_{k-1} \ldots u_{k-n_b}\right]^T \tag{13}$$

where the subscript k denotes the sample index.

The Frisch scheme provides the estimates of the measurement errors affecting the input and output signals u_k and y_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. 9 are assumed to be fixed in advance.

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

$$\left(\Sigma_\psi - \Sigma_{\tilde{\psi}}\right) \overline{\theta} = 0 \tag{14}$$

where the noise covariance matrix is given by

$$\Sigma_{\tilde{\psi}} = \begin{bmatrix} \sigma_y I_{n_a+1} & 0 \\ 0 & \sigma_u I_{n_b} \end{bmatrix} \tag{15}$$

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

$$\Sigma_{\tilde{\psi}} \approx \frac{1}{N} \sum_{k=1}^{N} \psi_k \psi_k^T \tag{16}$$

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. (14) can be solved, and $\overline{\theta}$ represents its solution.

The expression of Eq. (17) is determined:

$$\varepsilon_k(\overline{\theta}) = A(z^{-1})\, y_k - B(z^{-1})\, u_k \tag{17}$$

while the so-called sample autocovariance is defined in the form of Eq. (18):

$$r_{\varepsilon h, N} = \frac{1}{N}\sum_{l=1}^{N} \varepsilon_l(\overline{\theta})\, \varepsilon_{l+h}(\overline{\theta}) \tag{18}$$

where the subscript h in Eq. (18) indicates a time-shift.

The online control development requires a recursive estimate of the model parameters represented by the vector θ_k of Eq. (9), while the input and output data u_k and y_k acquired online 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. (14), (16) and (18) is expressed by means of an online scheme.

Note that the expressions of Eqs. (16) and (18) are required in their recursive form. Therefore, while the derivation of the online form of the covariance matrix update is easily obtained as in the form of Eq. (19):

$$\Sigma_{\tilde{\psi}k} = \frac{k-1}{k}\, \Sigma_{\tilde{\psi}k} + \frac{1}{k}\, \psi_k \psi_k^T \tag{19}$$

the formulation of the autocovariance expression $r_{\varepsilon h, k}$ can be obtained recursively for $1 \le l \le k$ only if the approximated expression of Eq. (20) is considered:

$$\varepsilon_l(\overline{\theta}_k) \approx \varepsilon_l(\overline{\theta}_l) \tag{20}$$

for $l < k$. In this way, only the residual $\varepsilon_k(\overline{\theta}_k)$ has to be computed at time step k using the lagged data in the vector ψ_k and the updated estimate $\overline{\theta}_k$ of the model parameters. The online computation of the expression of the autocovariance matrix of Eq. (21):

$$r_{\varepsilon h, k} = \frac{k-1}{k}\, r_{\varepsilon k, k-1} + \frac{1}{k}\, \varepsilon_k(\overline{\theta}_k)\, \varepsilon_{k+h}(\overline{\theta}_k) \tag{21}$$

can be achieved using only the vector $\varepsilon_{k+h}(\overline{\theta}_k)$ at each time step. 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 online, in order to cope with time-varying systems, this chapter 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. (19) and (21) with the approximation of Eq. (20) 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 autocovariance matrices in the form of Eq. (22):

$$\begin{cases} H_{\Sigma_{\tilde{\psi}_k}} = \omega(\delta)\,\Sigma_{\tilde{\psi}k} \\ h_{\varepsilon h,k} = \omega(\delta)\,r_{\varepsilon h,k} \end{cases} \tag{22}$$

where $\omega(\delta)$ is a scaling factor that coincides with k when no adaptation is introduced. In this way, the updated expressions have the form:

$$\begin{cases} H_{\Sigma_{\tilde{\psi}_k}} = (1-\delta)\,H_{\Sigma_{\tilde{\psi}_{k-1}}} + \delta\,\psi_k\,\psi_k^T \\ h_{\varepsilon h,k} = (1-\delta)\,h_{\varepsilon h,k-1} + \delta\,\varepsilon_k\big(\overline{\theta}_k\big)\,\varepsilon_{k+h}\big(\overline{\theta}_k\big) \end{cases} \tag{23}$$

with $0<\delta<1$ representing the forgetting factor. Thus, the adaptive Frisch scheme algorithm is implemented via Eq. (23) in three steps. First, θ_0, $\Sigma_{\tilde{\psi}0}$ and $r_{\varepsilon 0,h}$ with $h \leq n_a$ are initialised. Moreover, at each recursion step, by means of $r_{\varepsilon h,k}$, the noise variances σ_u and σ_y are computed. Finally, at each recursion step, $\overline{\theta}_k$ is determined by solving Eq. (14) via the expression of Eq. (23). In this way, the vector θ_k contains the estimates of the model parameter derived at the step k.

The results achieved by the online identification method recalled in this section were obtained in the Matlab® and Simulink® environments as summarised in Section 4.

Finally, once the parameters θ_k of the discrete-time linear time-varying model of the nonlinear dynamic process of Eq. (3) have been computed at each time step k, the adaptive controller is derived as summarised in Section 3.2.

3.1. Data-driven fuzzy controller derivation

This section describes the derivation of the fuzzy controller model. Once a reasonably accurate fuzzy description of the considered benchmark has been available, as described above, it is used offline to directly estimate the nonlinear fuzzy controllers. As already remarked, this design procedure differs from the approach proposed in Simani [11]. In fact, the control design proposed in this chapter relies on the so-called *model inverse control* principle, which is solved suing the fuzzy identification approach recalled above.

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 Simani [11], 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 Simani et al. [6] can be exploited, which is based on a parametrised search technique applied at a higher level to formulate the control objectives and constraints.

In this way, the estimated controller based on the inverse process 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. (8) has been derived for modelling the wind turbine dynamic process of Eq. (3) in its equivalent discrete-time form of Eq. (24):

$$y_{k+1} = f(x_k, u_k) \tag{24}$$

And, in particular, the TS fuzzy representation has the form of Eq. 25:

$$y_{k+1} = \frac{\sum_{i=1}^{M} \mu_i^{(m)}\left(x_k^{(m)}\right)\left(a_i^{(m)} x_k^{(m)} + b_i^{(m)}\right)}{\sum_{i=1}^{M} \mu_i^{(m)}\left(x_k^{(m)}\right)} \tag{25}$$

The current state $x_k = [y_k, \ldots, y_{k-n+1}, u_{k-1}, \ldots, u_{k-n+1}]^T$ and the input u_k represent the inputs that drive the model of Eq. (25). Its output represents the prediction of the system output at the next sample y_{k+1}. The model of Eq. (25) requires the estimated membership functions $\mu_i^{(m)}$, the state $x^{(m)}$ and the parameters $a_i^{(m)}$ and $b_i^{(m)}$ of the controlled system, which are denoted by the superscript (m).

Therefore, the input u_k generated by the control law feeds the monitored process such that its output y_{k+1} asymptotically follows the desired (reference) output r_{k+1}. This behaviour is obtained using the inverse model principle, represented by the expression of Eq. (26):

$$u_{k+1} = f^{-1}\left(x_k^c, r_k\right) \tag{26}$$

that is a nonlinear function of the vector x_k^c and the reference r_k.

However, in general, with reference to Eq. (26), it is difficult to determine the analytical expression of the inverse function $f^{-1}(\cdot)$. Therefore, the methodology proposed in this chapter suggested to exploit the identified fuzzy TS prototype of Eq. (25) to provide the particular state $x_k^{(m)}$ at each time step k. In this way, from this mapping, the inverse mapping $u_{k+1} = f^{-1}\left(x_k^{(c)}, r_k\right)$ is directly identified the form of Eq. (8), if the controlled system is stable, and in particular in the form of Eq. (27):

$$u_{k+1} = \frac{\sum_{i=1}^{M} \mu_i^{(c)}\left(x_k^{(c)}\right)\left(a_i^{(c)} x_k^{(c)} + b_i^{(c)}\right)}{\sum_{i=1}^{M} \mu_i^{(c)}\left(x_k^{(c)}\right)} \tag{27}$$

where the state $x_k^{(c)} = \left[x_k^{(m)}, r_{k-1}, \ldots, r_{k-n+1}\right]^T$ and the reference r_k signal represent the inputs of the identified controller model. The model of Eq. (27) contains the estimated membership functions $\mu_i^{(c)}$ and the parameters $a_i^{(c)}$ and $b_i^{(c)}$ of the identified controller model, which are denoted by the superscript (c).

In this way, the series connection between the controller model (i.e. the identified inverse process model) and the process model itself should lead to an identity mapping as in Eq. (28):

$$y_{k+1} = f\left(x_k^{(m)}, u_k\right) = f\left(x_k^{(m)}, f^{-1}\left(x_k^{(c)}, r_k\right)\right) = r_{k+1} \tag{28}$$

where $r_{k+1} = f\left(x_k^{(m)}, u_k\right)$ for a proper value of u_k. However, the expression of Eq. (28) holds in ideal conditions. However, 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 $|r_{k+1} - f\left(x_k^{(m)}, u_k\right)|$ 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 $a_i^{(c)}$ and $b_i^{(c)}$.

Moreover, the fuzzy model of the process is used for providing the state vector $x_k^{(m)}$. Therefore, the state of the fuzzy controller $x_k^{(c)}$ is updated using the process model state $x_k^{(m)}$ and the reference input r_k. These computations are performed using standard matrix operations, thus making the algorithm suitable for real-time implementations [12].

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_k and their references r_k. This mismatch can be compensated by means of the online mechanism described by the expressions of Eqs. (25) and (27). 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 will replace the baseline wind turbine regulator of Section 2.

3.2. Model-based adaptive controller derivation

This section describes the model-based adaptive control strategy used in connection with the online estimation scheme presented above. In more detail, with reference to the wind turbine system recalled in Section 2, adaptive controllers for processes of the second order ($n_a = n = 2$) are designed. Moreover, the considered adaptive controllers are based on the trapezoidal method of discretisation.

With reference to Eq. (9), the transfer function of the time-varying controlled system with $n_a = n_b = n = 2$ is considered, whose parameters estimated using the online identification approach recalled above:

$$\theta_k = \left[\widehat{a}_1, \widehat{a}_2, \widehat{b}_1, \widehat{b}_2\right]^T \tag{29}$$

Note that the subscript k for model and controller parameters will be dropped in order to simplify equations and formulas.

The control law corresponding to the discrete-time adaptive controller in its difference form of Eq. (30):

$$\begin{cases} \Delta e_k = e_k - e_{k-1} \\ u_k = K_p \left[\Delta e_k + \dfrac{T_s}{T_I} \dfrac{\Delta e_k}{2}\right] + u_{k-1} \end{cases} \tag{30}$$

with e_k representing the tracking error, with $e_k = r_k - y_k$, and r_k the reference (setpoint) signal. T_s is sampling time. The controller parameters K_p and T_I are here time-varying and derived from the online model parameters in the vector θ_k. The control law can be represented also in its feedback formulation as described by Eq. (31):

$$u_k = q_0 e_k + q_1 e_{k-1} + u_{k-1} \tag{31}$$

where the new controller variables q_0 and q_1 (or K_p and T_I) are derived from the relations of Eq. (32):

$$\begin{cases} q_0 = K_p \left(1 + \dfrac{T_s}{2T_I}\right) \\ q_1 = -K_p \left(1 - \dfrac{T_s}{2T_I}\right) \end{cases} \tag{32}$$

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:

$$K_p = 0.6 K_{P_u}, \qquad T_I = 0.5 T_u \tag{33}$$

that depend on the time-varying model parameters in the vector θ_k. In particular, when considering a second-order model described by its (time-varying) parameters $\widehat{a}_2$, $\widehat{a}_1$, $\widehat{b}_2$ and $\widehat{b}_1$, the variables K_{P_u} and T_u required by the Ziegler-Nichols method can be computed at each time step k from the following relations:

$$\begin{cases} K_{P_u} = \dfrac{\widehat{a}_1 - \widehat{a}_2 - 1}{\widehat{b}_2 - \widehat{b}_1} \\ T_u = \dfrac{2\pi T_s}{\arccos\gamma}, \qquad \textit{with } \gamma = \dfrac{\widehat{a}_2 \widehat{b}_1 - \widehat{a}_1 \widehat{b}_2}{2\widehat{b}_2} \end{cases} \tag{34}$$

In this way, the adaptive discrete-time linear controllers of Eq. (30) or (31) are designed on the basis of the time-varying linear model of Eq. (9) estimated via the online identification scheme from the data of the nonlinear wind turbine process of Eq. (3).

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. Simulation results

This section presents the simulation results achieved with the proposed data-driven and model methods relying on both the fuzzy modelling technique oriented to the identification of the fuzzy controller model and the adaptive control strategy using the online estimated models. The simulations achieved with these regulators are summarised in the following.

Regarding the fuzzy modelling and identification method, the GK clustering algorithm recalled in Section 3 exploits a number $M = 3$ of clusters and delays $n = 2$. These variables were applied for clustering the first data set consisting of $\{P_{gk}, \omega_{gk}, \beta_{rk}\}$. 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 $\{P_{gk}, \omega_{gk}, \tau_{gk}\}$. After this procedure, the structures of the TS prototypes were derived for each output y_k equal to P_{gk} and ω_{gk}. In this way, the 2 continuous-time outputs $y(t) = [\omega_g(t), \tau_g(t)]$ of the wind turbine continuous-time model of Eq. (3) are approximated by two TS fuzzy prototypes of Eq. (8).

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 [5]. In particular, the TS fuzzy model reconstructing the first output has a VAF index bigger than 90%, whereas 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 (multiple input single output) MISO fuzzy controller models with two inputs and one output have been used for the compensation of the blade pitch angle $\beta(t)$ and the generator torque $\tau_g(t)$. By using the inverse model principle, they were estimated exploiting the methodology recalled in Section 3.1. Again, the GK fuzzy clustering method has led to 2 fuzzy regulators applied to the data sets $\{\beta_{rk}, P_{gk}, \omega_{gk}\}$ and $\{\tau_{gk}, P_{gk}, \omega_{gk}\}$, 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 1** reports the

Recursive algorithm parameter	Value
$\overline{\theta}(0)$	$[0.1, 0.15, 0.20, 0.25\, 0.30, 0.35]^T$
$\Sigma_{\tilde{\psi}}(0)$	$10^{-1} I_7$
δ	0.995

Table 1. Initialisation parameters of the adaptive algorithm.

values of the percent Normalised Sum of Squared tracking Error (*NSSE%*) index defined in Eq. (35):

$$NSSE\% = 100\sqrt{\frac{\sum_{k=1}^{N}\left(r_k - y_k\right)^2}{\sum_{k=1}^{N} r_k^2}} \tag{35}$$

Noting that in partial load operation (region 1), the performance is represented by the comparison between the power produced by the generator, $y_k = P_{gk}$, with respect to the theoretical maximum power output, $r_k = P_r$. On the other hand, in full load operation (region 2), the tracking error is given by the difference between the generator speed, $y_k = \omega_{gk}$, and its nominal value, $r_k = \omega_{nom}$. The achieved results show the good properties of the designed fuzzy controllers, as represented also in **Figure 2**.

Figure 2 depicts the signal representing generator speed $\omega_g(t)$ in bold grey line with respect to its desired value ω_{nom} in dashed black line. It can be noted that in full load conditions, the fuzzy controllers derived via the data-driven approach lead to tracking errors smaller than the wind turbine baseline governor recalled in Section 2. In fact, as shown in **Figure 1**, the baseline regulator is working in the interval 2200*s* < *t* < 3300*s*. On the other hand, the fuzzy controllers are exploited during the interval 3300*s* < *t* < 4400*s*, when the tracking error is much lower.

With reference to the second model-based 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. (3) were approximated by two second-order time-varying MISO discrete-time models of Eq. (9) with two inputs and one output. Using these one 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 online. In particular, for each output, two second-order ($n_a = n_b = 2$) time-varying MISO prototypes were identified, and the adaptive regulator parameters in Eq. (30) or (31) were computed analytically at each time step *k*.

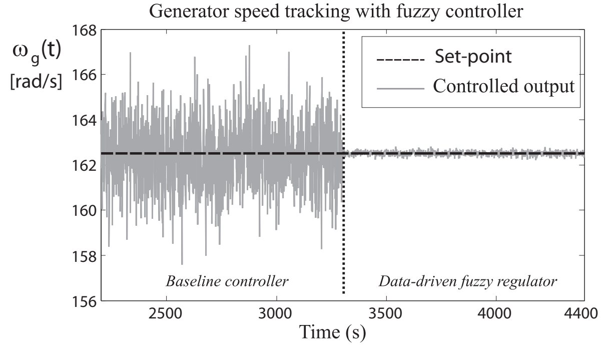

Figure 2. Generator speed (bold grey line) $\omega_g(t)$ and its reference (dashed black line) ω_{nom}.

Also in this case, with reference to the adaptive controller structure of Eq. (30) or (31), the parameters of the online controllers were tuned via the Ziegler-Nichols rules, applied to the LPV models. In this way, if both the model online parametric identification and the regulator tuning procedure are exploited, the parameter adaptation mechanisms should lead to good control performances.

The experiments with the adaptive regulators were simulated in the same situation of the fuzzy controllers. In this case, three online 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 runs with initial values for its parameters reported in **Table 1**.

With reference to the model-based adaptive approach, **Figure 3** depicts the setpoint $\omega_g(t)$ in bold grey line with respect to its desired value ω_{nom} in dashed black line. By considering the full load working conditions, the adaptive regulators have replaced the wind turbine baseline governor at $t \geq 3300s$.

Also for the case of the adaptive regulators, **Figure 3** highlights that the model-based approach leads to interesting performances.

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

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 the fuzzy regulators.

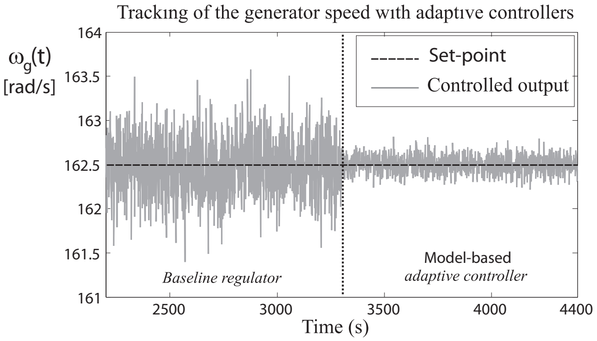

Figure 3. $\omega_g(t)$ tracking capabilities in full load conditions with adaptive controllers.

Controller type	Partial load (%)	Full load (%)
Fuzzy controller	37.17	17.85
Adaptive controller	28.73	13.67

Table 2. Controllers in partial and load operations: *NSSE*% values.

5. Conclusion

The chapter 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 are 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 chapter 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 were the practical instruments for assessing the most important characteristics of the developed control methodologies. The obtained results showed that the considered solutions represent viable, robust and reliable control applications to real wind turbine systems.

Author details

Silvio Simani[1]* and Paolo Castaldi[2]

*Address all correspondence to: silvio.simani@unife.it

1 Department of Engineering, University of Ferrara, Italy

2 Department of Engineering, University of Bologna, Italy

References

[1] Johnson KE, Pao LY, Balas MJ, Fingersh LJ. Control of variable–speed wind turbines: Standard and adaptive techniques for maximizing energy capture. IEEE Control Systems Magazine. 2006;**26**(3):70-81. DOI: 10.1109/MCS.2006.1636311

[2] Odgaard PF, Stoustrup J. A benchmark evaluation of fault tolerant wind turbine control concepts. IEEE Transactions on Control Systems Technology. 2015;**23**(3):1221-1228

[3] Zhao W, Stol K. Individual blade pitch for active yaw control of a horizontal–axis wind turbine. Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit; AIAA; Reno, NV, USA; 2007

[4] Juditsky A, Hjalmarsson H, Beneviste A, Delyon B, Ljung L, Sjöberg J, Zhang Q. Nonlinear black–box modelling in system identification: A mathematical foundation. Automatica. 1995;**31**(12):1691-1724

[5] Babuška R. Fuzzy Modeling for Control. Boston, USA: Kluwer Academic Publishers; 1998

[6] Simani S, Fantuzzi C, Rovatti R, Beghelli S. Parameter identification for piecewise linear fuzzy models in noisy environment. International Journal of Approximate Reasoning. 1999;**1**(22):149-167

[7] Landau Y. Adaptive Control. New York: Marcel Dekker; 1979. ISBN: 0-8247-6548-6

[8] Slotine JE, Li W. Applied Nonlinear Control. Prentice-Hall; 1991

[9] Simani S, Castaldi P. Data-driven and adaptive control applications to a wind turbine benchmark model. Control Engineering Practice. 2013;**21**(12):1678-1693. Special Issue Invited Paper. ISSN: 0967–0661. PII: S0967–0661(13)00155–X. DOI: http://dx.doi.org/10.1016/j.conengprac.2013.08.009

[10] Odgaard PF, Stoustrup J, Kinnaert M. Fault-tolerant control of wind turbines: A benchmark model. IEEE Transactions on Control Systems Technology. 2013;**21**(4):1168-1182. ISSN: 1063-6536. DOI: 10.1109/TCST.2013.2259235

[11] Simani S. Application of a data-driven fuzzy control design to a wind turbine benchmark model. Advances in Fuzzy Systems. 2012. (Web: http://www.hindawi.com/journals/afs/2012/504368/): 2012:1-12. Invited paper for the special issue: Fuzzy Logic Applications in Control Theory and Systems Biology (FLACE). ISSN: 1687–7101, e–ISSN: 1687-711X. DOI: 10.1155/2012/504368

[12] Rovatti R, Fantuzzi C, Simani S. High–speed DSP–based implementation of piecewise–affine and piecewise–quadratic fuzzy systems. Signal Processing Journal. 2000;**80**(6):951-963. Special Issue on Fuzzy Logic applied to Signal Processing. DOI: 10.1016/S0165-1684(00)00013-X