Chapter

Fault Diagnosis Techniques for a Wind Turbine System

Silvio Simani and Paolo Castaldi

Abstract

The fault diagnosis and prognosis of wind turbine systems represent a challenging issue, thus justifying the research topics developed in this work with application to safety-critical systems. Therefore, this chapter addresses these research issues and demonstrates viable techniques of fault diagnosis and condition monitoring. To this aim, the design of the so-called fault detector relies on its estimate, which involves data-driven methods, as they result effective methods for managing partial information of the system dynamics, together with errors, model-reality mismatch and disturbance effects. In particular, the considered data-driven strategies use fuzzy systems and neural networks, which are employed to establish non-linear dynamic links between measurements and faults. The selected prototypes are based on non-linear autoregressive with exogenous input descriptions, since they are able to approximate non-linear dynamic functions with arbitrary degree of accuracy. The capabilities of the designed fault diagnosis schemes are verified via a high-fidelity simulator, which describes the normal and the faulty behaviour of a wind turbine plant. Finally, the robustness and the reliability features of the proposed methods are validated in the presence of uncertainty and disturbance implemented in the wind turbine simulator.

Keywords: fault diagnosis, analytical redundancy, fuzzy prototypes, neural networks, diagnostic residuals, fault reconstruction, wind turbine simulator

1. Introduction

The increasing level of wind-generated energy in power generation worldwide also increases the levels of reliability and the so-called 'sustainability' shown by wind turbines. Wind turbine systems should generate the required amount of electrical power continuously, depending on the available wind speed, the grid's demand and possible malfunctions.

To achieve this aim, possible malfunctions affecting the process have to be properly detected and managed, before they degrade the nominal working conditions of the plant or become critical issues. Wind turbines with large rotors (i.e., of megawatt size) are very expensive systems, thus requiring an extremely high level of availability and reliability, in order to maximise the generated energy (at a reduced cost), with a minimisation of the operation and maintenance (O&M) services. In fact, the costs of the produced energy are mainly due to the installation cost of the wind turbine, while unplanned O&M costs could increase it up to about 30%, in particular when offshore installations are considered, see Odgaard [1].

These issues have motivated the development of fault diagnosis techniques that can be coupled with the fault-tolerant controllers (the so-called 'sustainable' systems). On the other hand, many turbine manufacturers adopt conservative approaches against faults, which lead to the shutdown of the plant in order to wait for O&M service. Hence, effective tools for coping with faults have to be investigated, in order to improve wind turbine features, particularly during faulty situations. This will lead to prevent critical failures that may affect other wind turbine components, thus avoiding unplanned replacement of functional parts, as well as the decrease of O&M costs, with the increase of the energy production. Moreover, the development of digital control systems, big data tools and artificial intelligence strategies enhance the development of new real-time condition monitoring, diagnosis and fault-tolerant control strategies for industrial processes, which can be available only on demand.

In recent years, many works have been proposed on the topics of fault diagnosis of wind turbines, as shown very recently in Habibi et al. [2] and Lan et al. [3]. Some of them are focused on the diagnosis of particular faults, for example, those affecting the drive-train system at a wind turbine level. Sometimes, these faults are better managed when the wind turbine system is considered in comparison to other parts of the whole plant, see Odgaard & Stoustrup [4]. Moreover, fault-tolerant control of wind turbines has been investigated, for example, in Parker et al. [5] and international cooperations on these problems were also proposed, see Odgaaard and Shafiei [6].

Fault diagnosis oriented to the sustainability feature when applied to safety-critical systems such as wind turbines has been proven to be a challenging issue, see Byrski and Byrski [7] and Xu et al. [8], thus motivating the research topics addressed in this chapter.

This point is fundamental as the increasing demand for energy generation using renewable sources has led to higher attention on renewable energy conversion systems, and in particular wind turbines. They represent very complex and safety-critical plants which require reliability, availability, maintainability and safety. Moreover, their efficiency to generate electrical power has to be maximised. This motivates novel research aspects, in particular in the context of diagnosis and control. The earlier diagnosis of faults and sustainable control solutions can lead to optimise energy conversion and guarantee the desired performances in the presence of possible malfunctions due to unexpected faults and disturbance.

Therefore, this chapter analyses the problem of the fault diagnosis for wind turbine systems, and the development of practical and reliable solutions for fault diagnosis, also known as fault detection and isolation (FDI). Further design of fault-tolerant controllers is not considered in this work, but it can rely on the tools considered in this chapter. In fact, the fault diagnosis module provides information on the faulty or fault-free conditions of the system, so that the controller activity can be compensated. This fault diagnosis task is enhanced by the use of fault estimators, which are obtained via data-driven approaches, as they offer effective tools for managing limited analytical knowledge of the process dynamics, together with noise and disturbance effects.

The first data-driven solution considered in this chapter uses fuzzy Takagi-Sugeno models, see Babuška [9], which are derived from a clustering algorithm, followed by an identification procedure, see Simani et al. [10]. A second solution is also considered, which relies on neural networks to describe the non-linear analytical links between measurement and fault signals. The chosen network architecture belongs to the Nonlinear AutoRegressive with eXogenous (NARX) input prototype, which can describe dynamic relationships along time. The training of the neural network fault estimators exploits standard training algorithm, that processes the data acquired from the process, see Roy and Ganguli [11].

The developed fault diagnosis strategies are verified by means of a high-fidelity simulator, which describes the normal and the faulty behaviour of a wind turbine plant. The achieved performances are verified in the presence of uncertainty and disturbance effects, thus validating the robustness features of the proposed schemes. The effectiveness verified from the achieved results suggests further investigations on more realistic applications of the proposed schemes.

The work is organised as follows. Section 2 recalls the wind turbine simulator. Section 3 illustrates the fault diagnosis methodologies relying on fuzzy and neural network prototypes. The obtained results are summarised in Section 4. Finally, Section 5 ends the chapter by outlining the key achievements of the study, and providing suggestions for future research issues.

2. Wind turbine simulator description

The wind turbine simulator used in this work was proposed in Odgaard et al. [12]. It describes the realistic behaviour of a three-blade horizontal-axis variable-speed pitch-controlled wind turbine coupled with a full converter generator. The overall system consists of four interconnected modules, that is, the wind driving process, the wind turbine, the measurement system and the baseline controller. The wind turbine block contains three submodels: the blade and the pitch system, the drive-train model and the generator system. The links between the system submodels are represented in **Figure 1**, with the fault diagnosis system to be designed. The simulator is able to generate several fault scenarios, see Odgaard et al. [12].

In the following, the description of these interconnected submodels is briefly recalled.

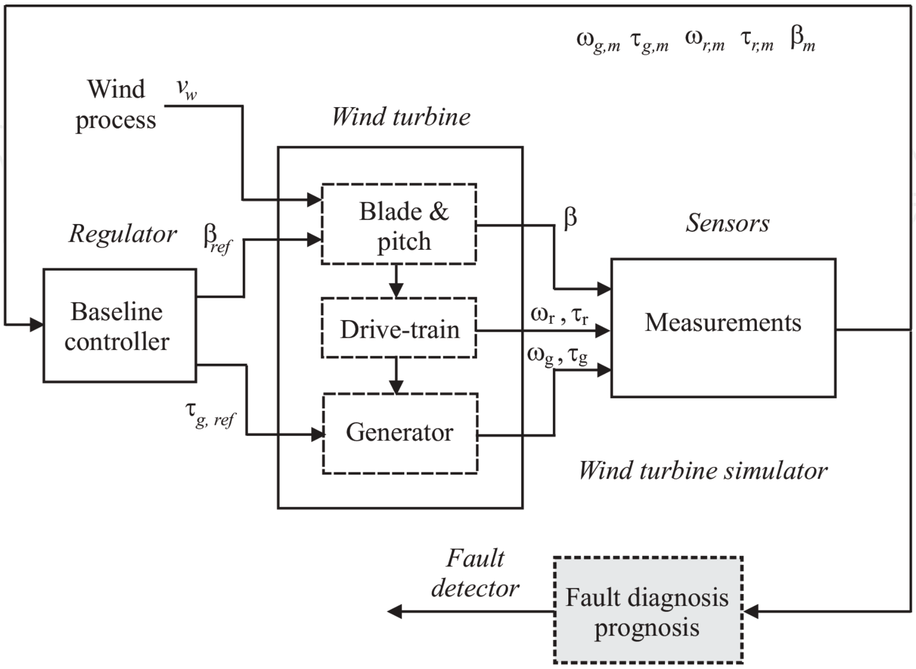

Figure 1.
The wind turbine simulator with its fault diagnosis system.

2.1 Wind turbine model

The turbine system consists of three submodels motivated by the power transmission flow. First, the blade and pitch block represents how the blades capture wind energy, which is based on the following aerodynamic law:

$$\tau_r(t) = \frac{\rho \pi R^3 C_q(\lambda(t), \beta(t)) v_w^2(t)}{2} \tag{1}$$

For each blade, Eq. (1) describes the torque acting on the rotor τ_r, depending on the squared wind speed v_w^2, the air density ρ and the rotor radius R. The coefficient C_q is usually defined using a two-dimensional map depending on the blade pitch angle β and the tip-speed ratio λ, that is, the ratio between the linear velocity of the blade tip and the wind speed. This map is represented by means of a look-up table. The blade and pitch system includes the dynamics of the pitch angle hydraulic piston servo system, which is approximated as a second-order transfer function of Eq. (2):

$$\frac{\beta(s)}{\beta_{ref}(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2}$$

where β_{ref} is the reference pitch angle computed by the turbine controller, while ζ and ω_n are the transfer function parameters.

The drive-train system determines the power flow through the gear box from the rotor toward the electric generator, whose dynamics are described as in Eq. (3):

$$\begin{cases} J_r\dot{\omega}_r = \tau_r - K_{dt}\theta_\Delta - (B_{dt} + B_r)\omega_r + \dfrac{B_{dt}}{N_g}\omega_g \\ J_g\dot{\omega}_g = \dfrac{\eta_{dt}K_{dt}}{N_g}\theta_\Delta + \dfrac{\eta_{dt}B_{dt}}{N_g}\omega_r - \left(\dfrac{\eta_{dt}B_{dt}}{N_g^2} + B_g\right)\omega_g - \tau_g \\ \dot{\theta}_\Delta = \omega_r - \dfrac{\omega_g}{N_g} \end{cases} \tag{3}$$

where J_r and J_g are the inertia moments of the rotor and generator shafts, respectively. K_{dt} is the torsion stiffness, B_{dt} is the torsion damping factor, B_g is the viscous friction of the generator shaft, B_r is the viscous friction of the low-speed shaft, N_g is the gear ratio, η_{dt} is the efficiency and θ_Δ is the torsion angle.

Finally, the generator submodel represents the converter dynamics by means of first-order transfer function of Eq. (4):

$$\frac{\tau_g(s)}{\tau_{g,ref}(s)} = \frac{\alpha_g}{s + \alpha_g} \tag{4}$$

where $\tau_{g,ref}$ is the reference torque defined by the controller and α_g is the transfer function parameter.

Finally, the generated power P_g is computed as the product of the generator torque by its speed, decreased by the efficiency coefficient η_g:

$$P_g = \eta_g\omega_g\tau_g \tag{5}$$

As sketched in **Figure 1**, the signals generated by the wind turbine system are assumed to be acquired through the measurement block, whose objective is to simulate the real behaviour of sensors and actuators. Therefore, the measured signals are modelled as sum of their actual value and white Gaussian process terms. Moreover, the wind turbine simulator includes a baseline controller, represented by a PID standard regulator, which regulates the generated power on the basis of the actual wind speed, as shown in Odgaard & Stoustrup [4] and Odgaard et al. [12].

2.2 Simulated fault scenario

The wind turbine simulator includes the generation of three different typical fault cases, that is, sensor, actuator and system faults, see Odgaard and Stoustrup [4] and Odgaard et al. [12].

For the case of the sensor faults, they are generated as additive signals on the affected measurements. As an example, the faulty sensor of faulty pitch angle β_m provides wrong measurements on blade orientation; thus, if not handled, the controller cannot fully track the power reference signal.

On the other hand, actuator faults lead to the alteration of pitch angle or the generator torque transfer functions of Eqs. (2) and (4), by modifying their dynamics. They simulate a pressure drop in the hydraulic circuit of the pitch actuator or an electronic break down in the converter device.

Finally, a system fault affects the drive train of the turbine, which is described as a slow variation in time of the friction coefficient. This can be due to the effect of wear and tear along time of the mechanical parts.

These nine fault cases are summarised in **Table 1**, which also highlights which measured signals are affected by them, as shown in **Figure 1**.

With these assumptions, the overall model of the wind turbine process can be represented as a non-linear continuous-time function $\mathbf{f}_{wt}$ describing the evolution of the turbine state vector $\mathbf{x}_{wt}$ excited by the input vector $\mathbf{u}$:

$$\begin{cases} \dot{\mathbf{x}}_{wt}(t) = \mathbf{f}_{wt}(\mathbf{x}_{wt}, \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{x}_{wt}(t) \end{cases} \tag{6}$$

where in this case, the state of the system is considered equal to the monitored system output, that is, the rotor speed, the generator speed and the generated power:

$$\mathbf{x}_{wt}(t) = \mathbf{y}(t) = \left[\omega_{g,m1}, \omega_{g,m2}, \omega_{r,m1}, \omega_{r,m2}, P_{g,m}\right]$$

Fault case	Fault type	Affected measurement
1	Sensor	$\beta_{1,m1}$
2	Sensor	$\beta_{2,m2}$
3	Sensor	$\beta_{3,m1}$
4	Sensor	$\omega_{r,m1}$
5	Sensor	$\omega_{r,m2}$ and $\omega_{g,m2}$
6	Actuator	Pitch system of blade #2
7	Actuator	Pitch system of blade #3
8	Actuator	$\tau_{g,m}$
9	System	Drive train

Table 1.
Wind turbine simulator fault scenario.

On the other hand, the input vector:

$$\mathbf{u}(t) = \left[\beta_{1,m1}, \beta_{1,m2}, \beta_{2,m1}, \beta_{2,m2}, \beta_{3,m1}, \beta_{3,m2}, \tau_{g,m}\right]$$

consists of the measurements of the pitch angles from the three redundant sensors, as well as the measured torque. These signals are sampled with sample time T in order to acquire a number N of data $\mathbf{u}(k)$, $\mathbf{y}(k)$ with $k = 1, \ldots, N$, in order to implement the data-driven fault diagnosis solutions proposed in this chapter.

3. Fault diagnosis techniques

This chapter considers two data-driven approaches, relying on fuzzy system and neural network structures, which are used to design the fault diagnosis schemes. Therefore, this section briefly introduces the general scheme of the fault diagnosis strategy, by recalling the basic features of the fuzzy systems and neural networks, as addressed in Sections 3.1 and 3.2, respectively. Moreover, these architectures, which are represented by NARX structures, are exploited residual generators for solving the problem of fault diagnosis, according to the analytical redundancy principle, see Chen and Patton [13].

In order to solve the fault diagnosis problem, this work assumes that the wind turbine system is affected by *equivalent* additive faults on the input and the output measurements, as well as measurement errors, as described by Eq. (7):

$$\begin{cases} \mathbf{u}(k) = \mathbf{u}^*(k) + \tilde{\mathbf{u}}(k) + \mathbf{f}_u(k) \\ \mathbf{y}(k) = \mathbf{y}^*(k) + \tilde{\mathbf{y}}(k) + \mathbf{f}_y(k) \end{cases} \tag{7}$$

where $\mathbf{u}^*(k)$ and $\mathbf{y}^*(k)$ represent the actual process variables, $\mathbf{u}(k)$ and $\mathbf{y}(k)$ are the measurements acquired from the sensors, while $\tilde{\mathbf{u}}(k)$ and $\tilde{\mathbf{y}}(k)$ describe the measurement errors. According to the description of Eq. (7), signals of the faults $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$ also have *equivalent* additive effects. Obviously, these functions are different from zero in faulty cases. In general, the vector $\mathbf{u}(k)$ has r components, that is, the number of process inputs, while $\mathbf{y}(k)$ has m elements, that is, the number of process outputs.

Among the possible approaches exploited for residual generation, and based on the analytical redundancy principle, this work proposes to exploit fuzzy system and neural network structures, which provide an on-line estimation $\hat{\mathbf{f}}(k)$ of the fault signals $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$. Hence, as shown in **Figure 1**, the so-called diagnostic residuals $\mathbf{r}(k)$ are equal to the estimated fault signals, $\hat{\mathbf{f}}(k)$, which are computed by the general fault estimator, as highlighted by Eq. (8):

$$\mathbf{r}(k) = \hat{\mathbf{f}}(k) \tag{8}$$

The variable $\hat{\mathbf{f}}(k)$ is the generic fault vector, that is, $\hat{\mathbf{f}}(k) = \left\{\hat{f}_1(k), \ldots, \hat{f}_{r+m}(k)\right\}$. Therefore, the general fault estimate $\hat{f}_i(k)$ can be equal to one of the i components of the fault vectors $\mathbf{f}_u(k)$ or $\mathbf{f}_y(k)$ in Eqs. (7), with $i = 1, \ldots, r + m$.

The residual generation scheme exploiting the fault estimators as residual generator is depicted in **Figure 2**. Note that this strategy is able to provide both the fault detection and isolation tasks, that is, the fault diagnosis function, see Chen and Patton [13].

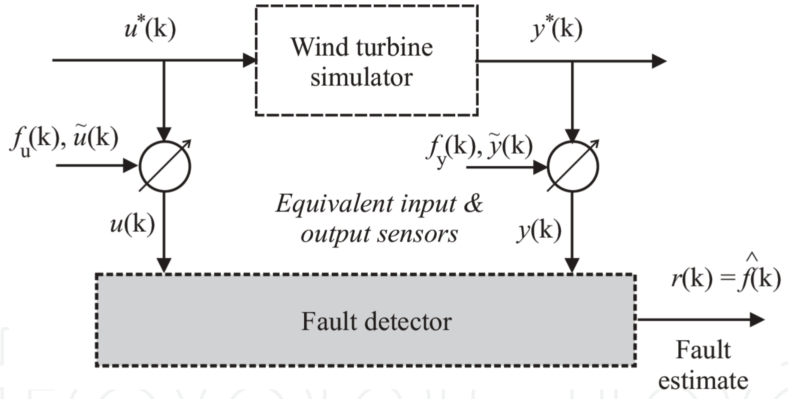

Figure 2.
Fault detectors for fault diagnosis.

Figure 2 shows that in general the residual generators use the acquired input and output measurements $\mathbf{u}(k)$ and $\mathbf{y}(k)$. As first step, the fault diagnosis scheme consists of the fault detection task. In this case, as the residual is equal to the estimated fault signal, it is easily performed via a proper thresholding logic directly operating on the residual itself, without requiring complex elaboration with proper evaluation functions, as shown in Chen and Patton [13]. Therefore, the occurrence of the ith fault can be simply detected via the threshold logic of Eq. (9) applied to the ith residual $r_i(k)$:

$$\begin{cases} \overline{r}_i - \delta\sigma_{r_i} \leq r_i \leq \overline{r}_i + \delta\sigma_i & \text{fault-free case} \\ r_i < \overline{r}_i - \delta\sigma_{r_i} \text{or } r_i > \overline{r}_i + \delta\sigma_{r_i} & \text{faulty case} \end{cases} \tag{9}$$

with $r_i(k)$ representing the ith component of the vector $\mathbf{r}(k)$. If it is considered as a random variable, its means $\overline{r}_i$ and variance $\sigma_{r_i}^2$ values can be estimated in fault-free condition, after the acquisition of N samples, according to Eq. (10):

$$\begin{cases} \overline{r}_i = \dfrac{1}{N}\sum\limits_{k=1}^{N} r_i(k) \\ \sigma_{r_i}^2 = \dfrac{1}{N}\sum\limits_{k=1}^{N} \left(r_i(k) - \overline{r}_i\right)^2 \end{cases} \tag{10}$$

Note that the parameter $\delta \geq 2$ represents a tolerance variable, which has to be properly tuned in order to effectively separate the fault-free from the faulty conditions. A common choice of δ can rely on the three-sigma rule, otherwise extensive simulations can be exploited for optimising this δ value, see Chen and Patton [13].

Once the fault detection phase is accomplished, the fault isolation task is directly obtained by means of a bank of estimators. As described by Eq. (7), the faults are considered as equivalent signals that are injected and affect the input measurements via the signal $\mathbf{f}_u$, or the output measurements by means of $\mathbf{f}_y$.

According to the scheme depicted in **Figure 3**, in order to uniquely isolate one of the input or output faults, under the assumption that multiple faults cannot occur, a bank of multi-input single-output (MISO) fault estimators is designed. In general, the number of this estimators is equal to the number of faults that have to be diagnosed, that is, which coincides with the number of input and output measurements, $r + m$. Therefore, the ith estimator providing the reconstruction of the fault $\hat{f}(k) = r_i(k)$ is driven by the components of the input and output signals $\mathbf{u}(k)$ and $\mathbf{y}(k)$, respectively. These components are selected in order to be sensitive to the

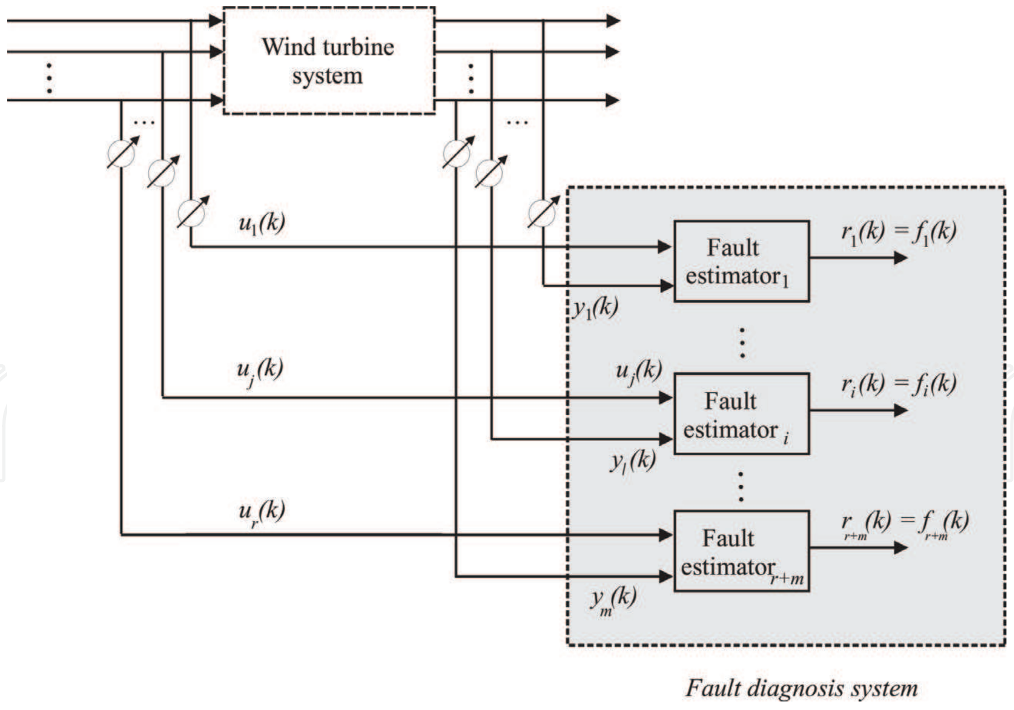

Figure 3.
Residual generators bank with $r_i(k) = f_i(k)$.

specific fault $f_i(k)$. In fact, the design of these fault estimators is enhanced by the fault sensitivity analysis described in Section 3.3. For each case, the fault modes and their resulting effects on the rest of the system are analysed, and in particular the most sensitive input $u_j(k)$ and output $y_l(k)$ measurements to that specific fault situation are selected. In this way, by means of the fuzzy system and neural network tools, it will be possible to derive the dynamic relationships between the input-output measurements, $u_j(k)$ and $y_l(k)$, and the faults $f_i(k)$, as highlighted by **Figure 3**.

Figure 3 shows this fault estimator bank, where the fault estimators are driven by the input-output signals selected via the fault sensitivity analysis procedure. In this way, the residual $r_i(k) = \hat{f}_i(k)$ is insensitive only to the fault affecting those inputs and outputs, $u_j(k)$ and $y_l(k)$, defined by the selector blocks. It is worth noting that, using this configuration, multiple faults occurring at the same time cannot be correctly isolated.

As already remarked, the sensitivity analysis, which has to be executed before the design of the fault estimators, suggests how to select the input-output signals feeding the fault estimator modules. After this selection procedure is performed, as described in Section 3.3, the design of the fuzzy or neural network models is achieved, as recalled in Sections 3.1 and 3.2, respectively. Finally, the threshold test logic of Eq. (9) allows the achievement of the fault diagnosis task.

3.1 Fuzzy system modelling and identification

This section describes the design of the fault estimators described by means of the Takagi-Sugeno (TS) prototypes, see Takagi and Sugeno [14]. Therefore, the unknown relationships between the selected measurements and the faults are described by fuzzy models, which consist of a number of rules. These rules connect the measured signals acquired from the system under diagnosis to its faults, described in form of IF⇒THEN relations, processed by a fuzzy inference system (FIS), see Babuška [9].

According to this approach, the approximation of non-linear multi-input single-output (MISO) systems can be achieved by the Takagi-Sugeno (TS) fuzzy reasoning, as described in Babuška [9]. The TS modelling approach proposed here, as addressed in Takagi and Sugeno [14], describes the consequents as deterministic functions $g_i(\cdot)$ of the inputs, while the antecedents remain fuzzy propositions.

The fuzzy rule of the FIS has the form of Eq. (11):

$$R_i : IF \quad (\text{fuzzy combination of inputs}) \quad THEN \quad \text{output} = g_i(\text{inputs}) \tag{11}$$

where i refers to the number of rules. The antecedents are combined by means of membership functions $\lambda_i(\mathbf{x})$ that take into account the logical connectives expressed by linguistic propositions. The rule consequent function $g_i(\cdot)$ is defined as parametric function in the affine form of Eq. (12):

$$g_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} + b_i \tag{12}$$

where $\mathbf{a}_i$ is the parameter vector, and b_i is a scalar offset, while $g_i(\mathbf{x})$ is the ith rule output. The number of rules is supposed equal the number of clusters n_C used for partitioning the data into regions where the relations $g_i(\cdot)$ hold, see Babuška [9]. Furthermore, the antecedent of each rule defines the degree of fulfilment for the corresponding consequent model, defined by the membership function $\lambda_i(\mathbf{x})$. Therefore, the global model is expressed as a fuzzy composition of parametric models $g_i(\mathbf{x})$.

The TS prototype takes the form of the expression of Eq. (13):

$$\hat{f} = \frac{\sum_{i=1}^{n_C} \lambda_i(\mathbf{x}) g_i(\mathbf{x})}{\sum_{i=1}^{n_C} \lambda_i(\mathbf{x})} \tag{13}$$

Using this fuzzy approach, in general, the fault $\hat{f}$ can be reconstructed from suitable data acquired from the system under diagnosis. In other words, the fault $\hat{f}$ is a weighted average of affine functions $g_i(\mathbf{x})$ of the input-output measurements, where the weights are the combined degree of fulfilment $\lambda_i(\mathbf{x})$ of the system inputs.

It is worth noting that the system under investigation corresponds to the wind turbine process described in Section 2, which has a dynamic behaviour. Therefore, the considered input vector $\mathbf{x}$ of the TS model of Eq. (13) contains the current as well as delayed samples of the system input and output signals.

Therefore, in order to include dynamics into the static relation of Eq. (11), the consequents are described as discrete-time linear AutoRegressive models with eXogenous input (ARX) of order o, in which the regressor vector has the form of Eq. (14):

$$\mathbf{x}(k) = [\ldots, y_l(k-1), \ldots, y_l(k-o), \ldots u_j(k), \ldots, u_j(k-o), \ldots]^T \tag{14}$$

where $u_l(\cdot)$ and $y_j(\cdot)$ are the components of the actual system input and output vectors $\mathbf{u}(k)$ and $\mathbf{y}(k)$ is selected via the fault sensitivity analysis tool of Section 3.3, and exploited in the scheme of **Figure 3**. The variable k represents the time step, with $k = 1, 2, \ldots, N$. The affine parameters associated to the ith model of the Eq. (12) are collected into the vector:

$$\mathbf{a}_i = \left[\alpha_1^{(i)}, \ldots, \alpha_o^{(i)}, \delta_1^{(i)}, \ldots, \delta_o^{(i)}\right]^T \tag{15}$$

where the $\alpha_j^{(i)}$ coefficients refer to the output samples, while $\delta_j^{(i)}$ are associated to the input ones.

A powerful approach to the design of the ith FIS as approximator for the system under diagnosis begins with the partitioning of the available data $\mathbf{u}(k)$ and $\mathbf{y}(k)$ of Eq. (7) into subsets, known as cluster. A cluster is defined as a set of data that are more similar to each other rather than to the members of another cluster. The similarity among data can be expressed in terms of their distance from a particular item, exploited as the cluster prototype. Fuzzy clustering provides an effective tool to obtain a partitioning of data in which the transitions among subsets are smooth, rather than abrupt. Moreover, fuzzy clustering assumes that the data of each cluster are characterised by an affine behaviour, which is indeed modelled by the relation of Eq. (12). Different clustering methods have been proposed in literature, see for example, more recent works Graaff and Engelbrecht [15] and Jun et al. [16].

With reference to this work, the design of the FIS is considered as a system identification problem from the noisy data of Eqs. (7). In fact, the estimation of the consequent parameters $\mathbf{a}_i$ and b_i of Eq. (12) is required using the input-output data for designing the bank of the fault estimations reported in **Figure 3**. Moreover, the data are acquired from the measurements selected from the procedure suggested in Section 3.3. The identification scheme exploited in this work was proposed by the authors in Fantuzzi et al. [17]. This approach is based on the minimisation of the prediction errors of the individual TS local affine models considered as n_C-independent estimation problems. Their solutions rely on the estimation of errors-in-variables models in Fantuzzi et al. [17], which is also the assumption represented by Eq. (7).

Another key aspect, which is not considered here, regards the determination of the optimal number of clusters n_C, as the clustering algorithm assumes that the number of clusters n_C has been fixed. These issues are considered in the development of the estimation procedure properly integrated by the authors, which also determines the antecedent degrees of fulfilment μ_{ik} required by Eq. (13) and solved with curve fitting methods, see Babuška [9].

3.2 Neural network modelling and training

This study proposes a different data-driven approach, based on neural networks, which is exploited to implement the fault diagnosis block. This section briefly recalls their general structure and properties, which are used to implement the fault estimators.

Therefore, according to the scheme shown in **Figure 4**, a bank of neural networks is realised in order to reproduce the behaviour of the faults affecting the system under diagnosis using a proper set of input and output measurements. The neural network structure consists of different layers of neurons, also known as *perceptron,* see Haykin [18], modelled as a static function f. This function is described by an *activation function* with multiple inputs properly weighted by unknown parameters that determine the learning capabilities of the whole network.

A categorisation of these learning structures concerns the way in which their neurons are connected to each other, see Xu et al. [19]. This work proposes to use a feedforward network, also called multilayer perceptron, where the neurons are grouped into unidirectional layers. The first of them, the input layer, is directly fed by the network inputs; then, a hidden layer takes the inputs from the neurons of the input layer and transmits them the output to the neurons of the third layer, the output layer, which produces the final network outputs. According to this structure, neurons are connected from one layer to the next, but not within the same layer. The only constraint is the number of neurons in the output layer, that has to be equal to the number of actual network outputs. On the other hand, recurrent

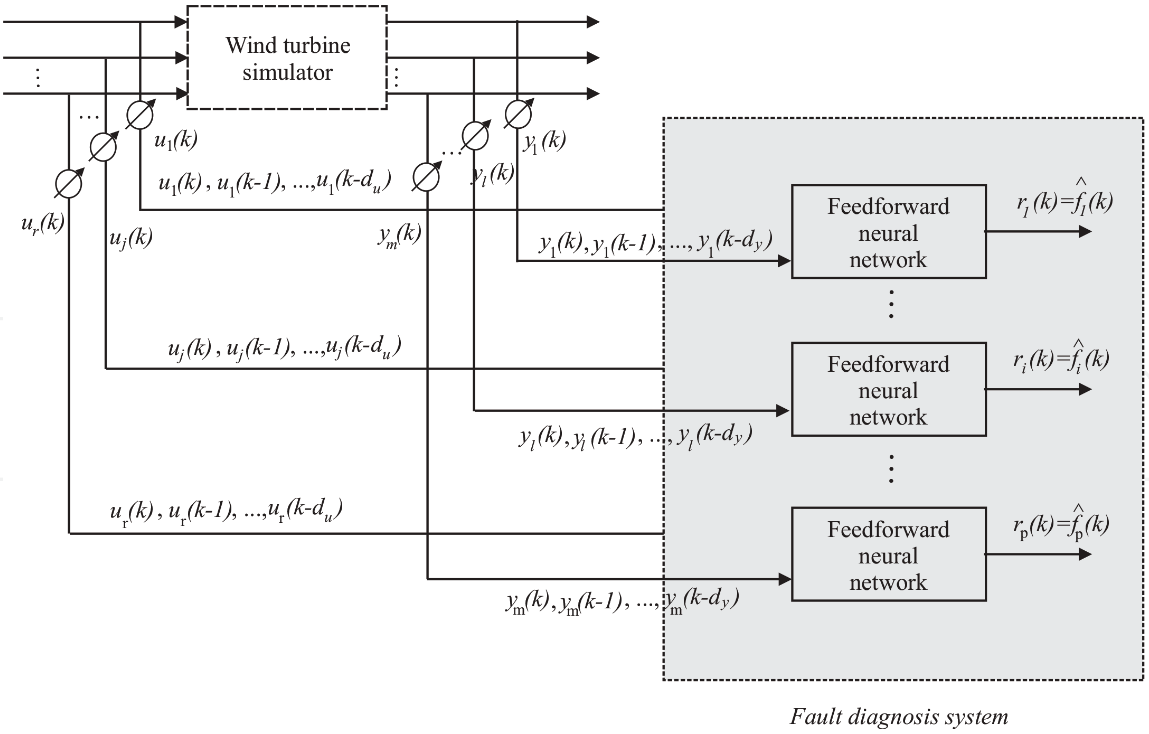
Wind turbine simulator
$u_1(k)$
$u_1(k)$, $u_1(k-1)$, ..., $u_1(k-d_u)$
$u_r(k)$
$u_j(k)$
$y_1(k)$
$y_l(k)$
$y_m(k)$
$y_1(k)$, $y_1(k-1)$, ..., $y_1(k-d_y)$
Feedforward neural network
$r_l(k)=\hat{f}_l(k)$
$u_j(k)$, $u_j(k-1)$, ..., $u_j(k-d_u)$
Feedforward neural network
$r_i(k)=\hat{f}_i(k)$
$y_l(k)$, $y_l(k-1)$, ..., $y_l(k-d_y)$
$u_r(k)$, $u_r(k-1)$, ..., $u_r(k-d_u)$
Feedforward neural network
$r_p(k)=\hat{f}_p(k)$
$y_m(k)$, $y_m(k-1)$, ..., $y_m(k-d_y)$
Fault diagnosis system

Figure 4.
Neural networks as fault estimators with $\hat{f}_i(k) = r_i(k)$.

networks are multilayer networks, in which the output of some neurons is fed back to neurons belonging to previous layers, thus the information flow in forward as well as in backward directions, allowing a dynamic memory inside the network, see Hunt et al. [20].

A noteworthy intermediate solution is provided by the multilayer perceptron with a tapped delay line, which is a feedforward network whose inputs come from a delay line. This study proposes to use this solution, defined as quasistatic neural network, as it represents a suitable tool to predict dynamic relationships between the input-output measurements and the considered fault functions. In this way, another NARX description is obtained, since the non-linear (static) network is fed by the delayed samples of the system inputs and outputs selected by the fault sensitivity analysis tool described in Section 3.3. Indeed, if properly trained, the NARX network can estimate the current (and the next) fault samples $f_j(k)$ on the basis of the selected past measurements of system inputs and outputs $u_l(k)$ and $y_j(k)$, respectively, in the same way of the fuzzy systems.

Therefore, with reference to the ith residual generator of **Figure 4**, which is used to design the estimator bank of **Figure 3**, this NARX network is described by the relation of Eq. (16):

$$\hat{f}_i(k) = F\left(\dots, u_j(k), \dots, u_j(k-d_u), \dots y_l(k-1), \dots, y_l\left(k-d_y\right), \dots\right) \tag{16}$$

where $\hat{f}_i(k)$ is the estimation of the generic ith fault, while $u_j(\cdot)$ and $y_l(\cdot)$ are the generic jth and lth components of the measured inputs and outputs $\mathbf{u}$ and $\mathbf{y}$, respectively, that are selected via the fault sensitivity analysis tool. k is the time step, d_u and d_y are the number of delay of inputs and outputs, respectively, which have to be properly selected. $F(\cdot)$ is the function realised by the static neural network, which depends on the layer architecture, the number of neurons, their weights and their activation functions. The NARX network used as generic fault $f_i(k)$ estimator is depicted in **Figure 4**.

The design parameters are represented by the number of neurons and the number of delays of the network inputs and outputs, while the value of the weights of each neuron are derived from the network training from the data acquired from the system under diagnosis, see Hunt et al. [20].

3.3 Fault sensitivity analysis

The design of the fault diagnosis schemes proposed for the application example considered in this chapter have been summarised in Section 4. However, the tool addressed in this chapter enhances the design of the banks of these fault estimators depicted in **Figure 3**.

This tool consists of a fault sensitivity analysis that has to be performed on the wind turbine simulator. It is aimed at defining the most sensitive measurements $u_j(k)$ and $y_l(k)$ with respect to the fault conditions $f_i(k)$ considered in Section 2.2. In practice, the considered fault signals have been injected into the wind turbine simulator, assuming that only a single fault may occur. Then, the relative mean square errors (RMSE) between the fault-free and faulty measured signals are evaluated, so that, for each fault, the most sensitive signal $u_j(k)$ and $y_l(k)$ can be selected. The results of the fault sensitivity analysis are summarised in **Table 2** for the wind turbine system.

In particular, the fault sensitivity analysis is conducted on the basis of a selection algorithm that is performed by introducing the normalised sensitivity function N_x, defined in Eq. 17:

$$N_x = \frac{S_x}{S_x^*} \tag{17}$$

with

$$S_x = \frac{\|x_f(k) - x_n(k)\|_2}{\|x_n(k)\|_2} \tag{18}$$

and

$$S_x^* = \max \frac{\|x_f(k) - x_n(k)\|_2}{\|x_n(k)\|_2} \tag{19}$$

The value of N_x indicates the effect of the considered fault case with respect to the general measured signal $x(k)$, with $k = 1, 2, \ldots, N$. The subscripts 'f' and 'n' indicate the faulty and the fault-free case, respectively. Therefore, the

Fault f_i	1	2	3	4	5
Measurements u_j, y_l	$\beta_{1,m1}$	$\beta_{2,m2}$	$\beta_{3,m1}$	$\omega_{r,m1}$	$\omega_{r,m1}$
RMSE	11.29	0.98	2.48	1.44	1.45
Fault f_i	6	7	8	9	
Measurements u_j or y_l	$\beta_{2,m1}$	$\beta_{3,m2}$	$\tau_{g,m}$	$\omega_{g,m1}$	
RMSE	0.80	0.73	0.84	0.77	

Table 2.
Fault sensitivity $f_i(k)$ with respect $u_j(k)$ and $y_l(k)$.

Fault case f_i	Most sensitive inputs u_j	Most sensitive outputs y_l
1	$\beta_{1,m1}, \beta_{1,m2}$	$\omega_{g,m2}$
2	$\beta_{1,m2}, \beta_{2,m2}$	$\omega_{g,m2}$
3	$\beta_{1,m2}, \beta_{3,m1}$	$\omega_{g,m2}$
4	$\beta_{1,m2}$	$\omega_{g,m2}, \omega_{r,m1}$
5	$\beta_{1,m2}$	$\omega_{g,m2}, \omega_{r,m2}$
6	$\beta_{1,m2}, \beta_{2,m1}$	$\omega_{g,m2}$
7	$\beta_{1,m2}, \beta_{3,m2}$	$\omega_{g,m2}$
8	$\beta_{1,m2}, \tau_{g,m}$	$\omega_{g,m2}$
9	$\beta_{1,m2}$	$\omega_{g,m1}, \omega_{g,m2}$

Table 3.
Fault sensitivity test.

measurements that are most affected by the considered fault lead to a value of N_x equal to 1. Otherwise, a smaller value of N_x, that is, close to zero, represents a signal $x(k)$ not affected by the fault. Those signals characterised by high value of N_x are thus selected as the most sensitive measurements, and they will be considered in the design of the fault diagnosis modules of the bank sketched in **Figure 3**.

The complete results of the fault sensitivity analysis are summarised in **Table 3**. For each fault case, the selected signals of the wind turbine benchmark are marked as inputs or outputs.

This method represents a key feature of the proposed approach to fault diagnosis. In fact, the fault estimators of the bank of **Figure 3** can be designed by exploiting a reduced number of signals, thus leading to a noteworthy simplification of the overall complexity, and a decrease in the computational cost of the training and identification phases.

4. Simulation results

This section summarises the simulations performed with the considered wind turbine benchmark, and the performances of the proposed fault diagnosis solutions. Due to the presence of the uncertainty and disturbance effects included in the benchmark, the robustness features of the developed fault diagnosis techniques are also verified in simulation.

With reference to the wind turbine benchmark of Section 2, all simulations are driven by the same wind mean speed sequence. It was acquired from a real measurement of wind speed, which represents a good coverage of typical operating conditions, as it ranges from 5 to 20 m/s, with a few spikes at 25 m/s, see Odgaard et al. [12]. The simulations last for 4400 s, with single fault occurrences. The discrete-time simulator runs at a sampling frequency of 100 Hz, so that $N = 440{,}000$ samples are acquired during each simulation. With reference to the different fault cases reported in Section 2.2, **Table 4** shows the shape and the timing of the fault modes affecting the process. They model input (actuator) or output (sensor) additive faults, which are used for sensitivity analysis of Section 3.3.

As an example, in order to highlight the actual fault effect on the wind turbine measurements, **Figure 5** shows the fault sensitivity test. In particular, the cases of the faults 1, 2, 3 and 8 in fault-free and faulty conditions are depicted.

Fault case	Fault type	Fault shape	Occurrence (s)
1	Actuator	Step	2000–2100
2	Actuator	Step	2300–2400
3	Actuator	Step	2600–2700
4	Actuator	Step	1500–1600
5	Actuator	Step	1000–1100
6	Sensor	Step	2900–3000
7	Sensor	Trapezoidal	3500–3600
8	Sensor	Step	3800–3900
9	Sensor	Step	4100–4300

Table 4.
Wind turbine simulator fault conditions.

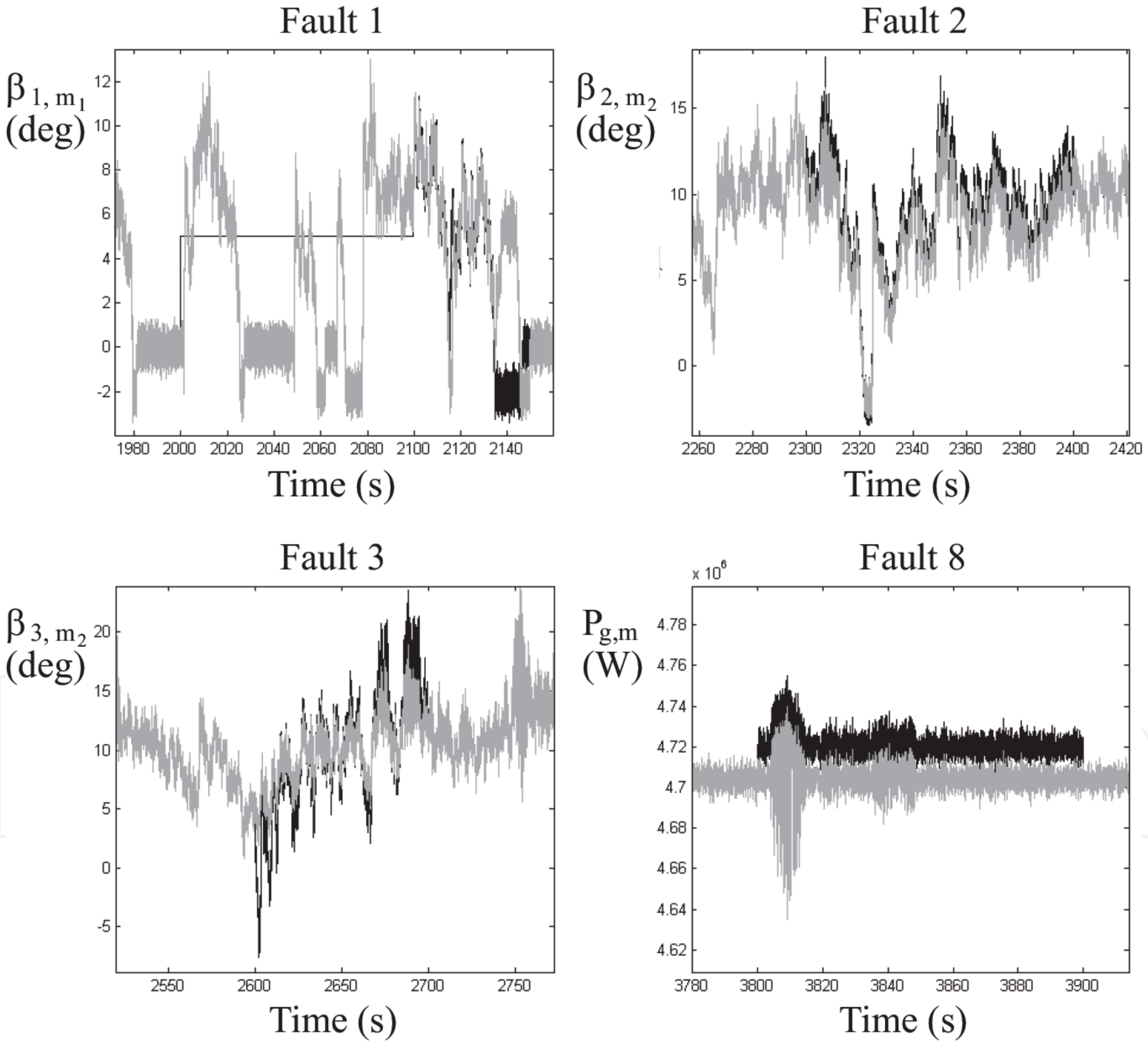

Figure 5.
Example of fault-free (grey line) and faulty (black line) signals.

4.1 Fuzzy estimators for fault diagnosis

The problem of the fault diagnosis of the wind turbine simulator is solved in this work by designing fuzzy prototypes as fault reconstructors. The considered approach is different from the one presented in Simani et al. [21], where the fuzzy models were used as output predictors.

Section 3.1 suggested to exploit the fuzzy c-means clustering algorithm. When applied to the data of the wind turbine simulator, a number $n_C = 4$ of clusters and $o = 3$ delays on input and output regressors were determined. The tool also generated the membership function points that are fitted through Gaussian membership functions. After data clustering, the regressands $\alpha_j^{(i)}$ and $\delta_j^{(i)}$ of Eq. (15) were identified for each cluster by following the procedure of Section 3.1. The TS models of Eq. (13) were thus implemented and nine fault estimators were designed, built and organised into the estimator scheme in order to accomplish the fault diagnosis task, as sketched in **Figure 3**.

The effectiveness of the fuzzy TS fault estimators used was assessed in terms of root mean squared error (RMSE), which is computed as the difference between the predicted $\hat{f}_i(k)$ and the actual fault $f_i(k)$ signals for each of the fuzzy estimators, with $i = 1, \ldots, 9$. **Table 5** summarises the achieved performance of the nine fault estimators of **Figure 3**.

In this case, these estimated signals $\hat{f}_i$ are directly exploited as diagnostic residuals r_i, as remarked by Eq. (8). They can be compared with the thresholds of Eq. (9), optimally selected in order to achieve the optimisation of the overall fault diagnosis performance indices, in terms of missed fault and the false alarm rates, see Ding [22]. In particular, **Table 6** summarises the values of the parameter δ of Eq. (9) for each fault estimator i.

Note that, in general, each of the nine fuzzy fault estimators described by the relations of Eqs. (13) and (14) has three inputs (see **Table 3**), with a number of delays $n = 3$ and $n_C = 4$ clusters. Therefore, the number of estimated parameters for each fuzzy MISO model (three inputs and one output) is equal to $(3 + 1) \times n = 12$. Moreover, for each fault estimator, the estimation of the fuzzy membership functions $\lambda_i(\cdot)$ of Eq. (13) with $i = 1, \ldots, n_C$ was required.

In the following, the main simulation results are summarised. Two actuator faults f_u and two sensor fault f_y are considered, namely the fault cases 1, 4, 8 and 9 of the scenarios recalled in Section 2.2.

According to **Table 3**, these faults caused the alteration of the monitored input and output signal **u**, **y** affecting the residual $r_1 = \hat{f}_1$, $r_4 = \hat{f}_4$, $r_8 = \hat{f}_8$ and $r_9 = \hat{f}_9$ generated by the fuzzy fault estimators. These faults $\hat{f}_i$ depicted in **Figure 6** demonstrate

Fault estimator $\hat{f}_i$	1	2	3	4	5
RMSE	0.016	0.023	0.021	0.020	0.019
Fault estimator $\hat{f}_i$	6	7	8	9	
RMSE	0.021	0.017	0.021	0.019	

Table 5.
Fuzzy fault estimator capabilities with RMSE.

Residual $r_i(k)$	1	2	3	4	5	6	7	8	9
δ	3.8	4.3	4.2	4.5	3.7	4.4	4.3	3.5	3.9

Table 6.
The parameter δ for the threshold selection.

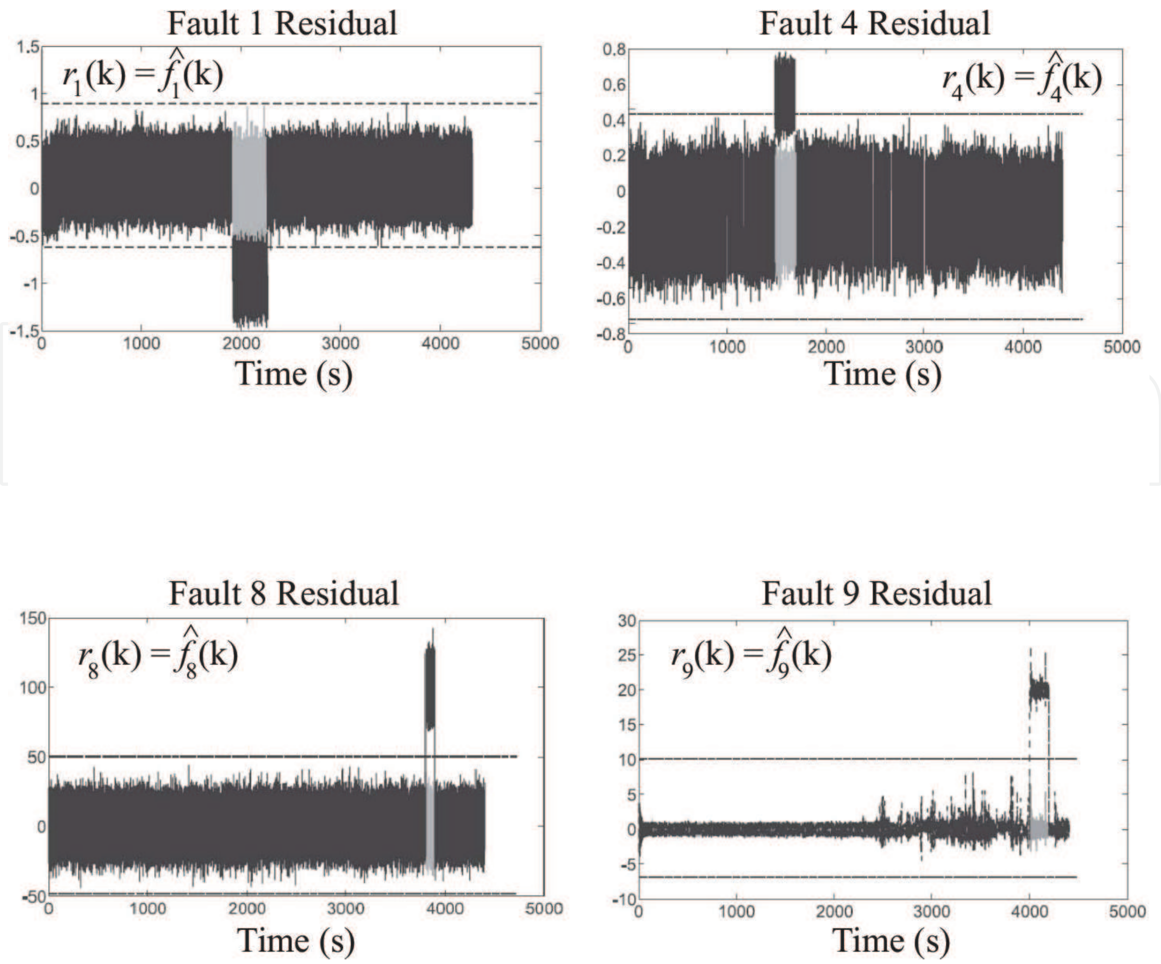

Figure 6.
Fault-free (grey line) and faulty (black continuous line) residuals with faults 1, 4, 8 and 9.

the achievement of the fault diagnosis task, as they exceed the threshold levels only when the relative fault is active, as recalled in **Table 4**.

Figure 6 depicts the reconstructed fault functions $\hat{f}_i(k)$ generated by the fuzzy estimators in faulty conditions (black continuous line) with respect to the fault-free residuals (grey line). The fixed thresholds are depicted with dotted lines. The considered residuals refer to the fault cases 1, 4, 8 and 9. It is worth noting that in fault-free conditions, the estimated fault functions $\hat{f}_i(k)$ are not zero due to both the model-reality mismatch. **Figure 6** also highlights the robustness and reliability features of the developed fuzzy estimators.

4.2 Neural networks for fault diagnosis

As for the fuzzy systems, nine NARX neural networks described in Section 3.2 were designed to estimate the nine faults affecting the acquired measurements, according to the scheme of **Figure 3**. The neural networks selected for fault diagnosis purpose consist of 3 layers, with 3 neurons in the input layer, 16 in the hidden one, and 1 neuron in the output layer. A number of $d_u = d_y = 4$ delays were selected in the relation of Eq. (16). Both the input and the hidden layers used sigmoidal activation functions, while the output layer exploits the linear one. According to **Table 3** and **Figure 4**, each of the nine neural networks was driven by three inputs.

As for the fuzzy models, the prediction efficacy of the designed neural networks was verified in terms of RMSE. The achieved results are summarised in **Table 7**, which were obtained by comparing the estimated faults with respect to the simulated ones.

Fault estimate $\hat{f}_i(\boldsymbol{k})$	1	2	3	4	5
RMSE	0.009	0.009	0.009	0.012	0.011
Fault estimate $\hat{f}_i(\boldsymbol{k})$	6	7	8	9	
RMSE	0.011	0.009	0.009	0.014	

Table 7.
Neural network performances.

$r_i(\boldsymbol{k})$	1	2	3	4	5	6	7	8	9
δ	4.2	4.9	4.7	5.1	4.2	4.6	4.8	4.1	4.3

Table 8.
δ values for the threshold selector.

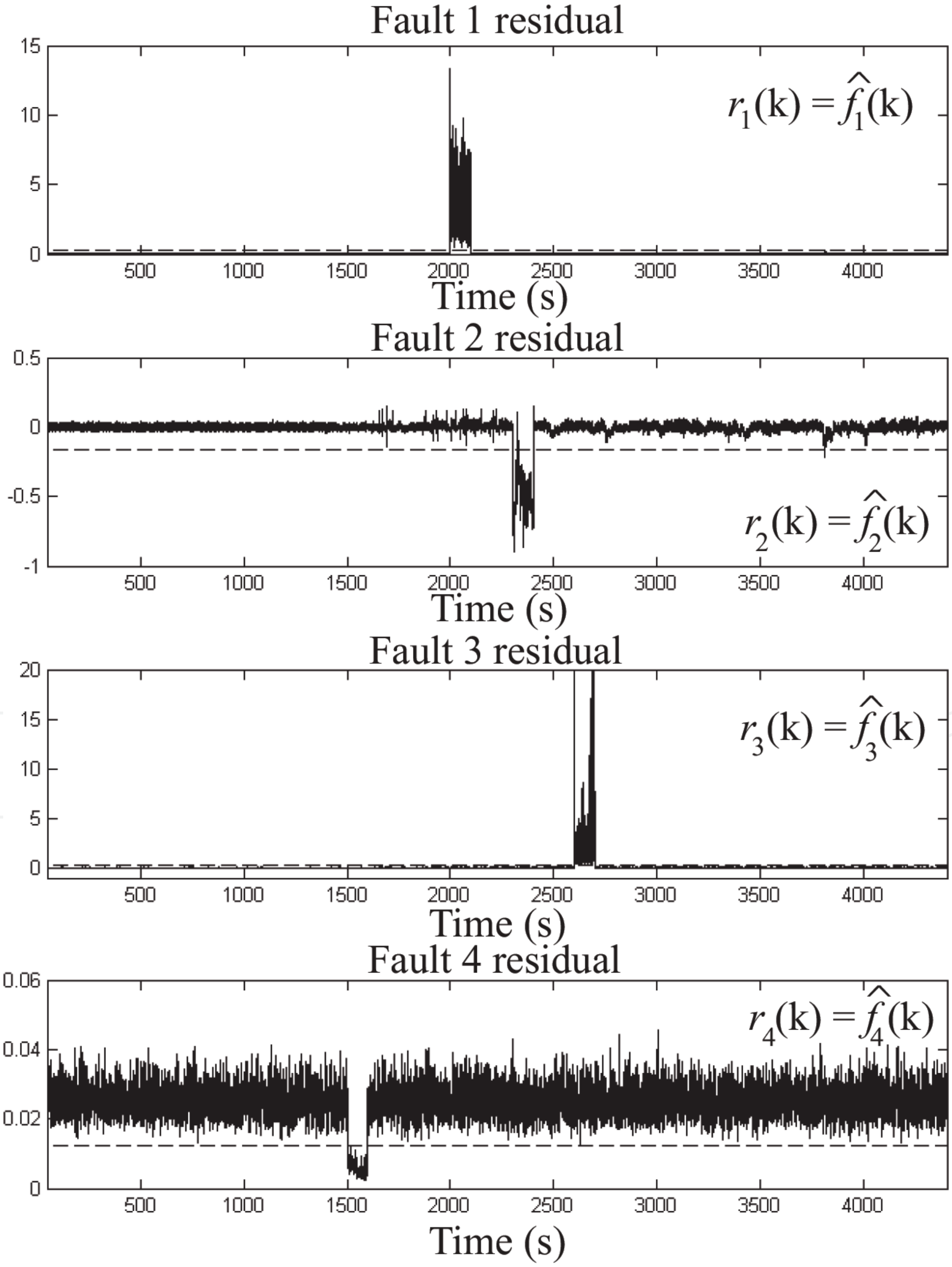

Figure 7.
Estimated faults (continuous line) $\hat{f}_i(k)$ and thresholds (dashed line) for cases 1, 2, 3 and 4.

The fault diagnosis task is thus achieved by comparing the residuals $r_i = \hat{f}_i(k)$ of Eq. (8) with fixed optimised thresholds, as described by Eq. (9). As for the fuzzy estimators, the values of the parameter δ of Eq. (9) for each fault estimator i is summarised in **Table 8**.

On the other hand, **Figure 7** shows an example of residual signals for the fault cases 1, 2, 3 and 4, together with the selected thresholds.

In particular, **Figure 6** depicts the residuals $\hat{f}_i(k)$ generated in faulty conditions by the neural network estimators (continuous line) compared with the fixed thresholds (dashed line). The considered residuals refer to the faults $f_1(k)$, $f_2(k)$, $f_3(k)$ and $f_4(k)$ of **Table 4**.

The achieved results show the effectiveness of the proposed fault diagnosis solutions, also with respect to disturbance and uncertainty effects on the wind turbine simulator, thus highlighting their potential application to real wind turbine systems.

5. Conclusion

The chapter studied data-driven tools for solving the problem of the fault diagnosis and prognosis of a wind turbine process. The design of this fault detector is based on the estimate of the fault itself, achieved by means of artificial intelligence methods. They were considered since these viable tools demonstrated to be able to cope with poor information on the process dynamics, in the presence of errors, model-reality mismatch and disturbance effects. In particular, these methodologies rely on fuzzy and neural network structures used to determine the non-linear dynamic links between measurements and fault signals. The selected structures belong to the non-linear autoregressive with exogenous input architectures, since they may model any non-linear dynamic relationship with arbitrary degree of accuracy. The fault diagnosis and prognosis strategies were validated via a high-fidelity simulator of a wind turbine process. The achieved performances in terms of reliability and robustness were thus tested by considering the presence of uncertainty and disturbance effects modelled by this wind turbine simulator. Further works will verify the features of the same fault diagnosis schemes when applied to real plants.

Author details

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

1 Department of Engineering, University of Ferrara, Italy

2 Department of Engineering, University of Bologna, Italy

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

References

[1] Odgaard PF. FDI/FTC wind turbine benchmark modelling. In: Patton RJ, editor. Workshop on Sustainable Control of Offshore Wind Turbines. Vol. 1. Hull, UK: Centre for Adaptive Science & Sustainability, University of Hull; 2012

[2] Habibi H, Nohooji HR, Howard I. Adaptive PID control of wind turbines for power regulation with unknown control direction and actuator faults. IEEE Access. 2018;**6**: 37464-37479. DOI: 10.1109/ACCESS.2018.2853090

[3] Lan J, Patton RJ, Zhu X. Fault–tolerant wind turbine pitch control using adaptive sliding mode estimation. Renewable Energy. 2018;**116**(Part B): 219-231. DOI: 10.1016/j.renene.2016.12.005

[4] 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

[5] Parker MA, Chong HN, Ran L. Fault-tolerant control for a modular generator–converter scheme for direct–drive wind turbines. IEEE Transactions on Industrial Electronics. 2011;**58**(1): 305-315

[6] Odgaaard PF, Shafiei SE. Evaluation of wind farm controller based fault detection and isolation. In: Proceedings of the IFAC SAFEPROCESS Symposium 2015. Vol. 48. IFAC; Paris, France: Elsevier; 2015. pp. 1084-1089. DOI: 10.1016/j.ifacol.2015.09.671

[7] Byrski J, Byrski W. A double window state observer for detection and isolation of abrupt changes in parameters. International Journal of Applied Mathematics and Computer Science. 2016;**3**(26):585-602. DOI: 10.1515/amcs-2016-0041

[8] Xu F, Puig V, Ocampo-Martinez C, Olaru S, Niculescu SI. Robust MPC for actuator–fault tolerance using set-based passive fault detection and active fault isolation. International Journal of Applied Mathematics and Computer Science. 2017;**27**(1):43-61. DOI: 10.1515/amcs-2017-0004

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

[10] 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. Elsevier

[11] Roy N, Ganguli R. Filter design using radial basis function neural network and genetic algorithm for improved operational health monitoring. Applied Soft Computing Journal. 2006;**6**(2):154-169

[12] 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

[13] Chen J, Patton RJ. Robust Model–Based Fault Diagnosis for Dynamic Systems. Boston, MA, USA: Kluwer Academic Publishers; 1999

[14] Takagi T, Sugeno M. Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics. 1985;**SMC-15**(1): 116-132

[15] Graaff AJ, Engelbrecht AP. Clustering data in stationary environments with a local network neighbourhood artificial immune

system. International Journal of Machine Learning and Cybernetics. 2012;**3**(1):1-26. DOI: 10.1007/s13042-011-0041-0

[16] Jun W, Shitong W, Chung F-L. Positive and negative fuzzy rule system, extreme learning machine and image classification. International Journal of Machine Learning and Cybernetics. 2011;**2**(4):261-271. DOI: 10.1007/s13042-011-0024-1

[17] Fantuzzi C, Simani S, Beghelli S, Rovatti R. Identification of piecewise affine models in noisy environment. International Journal of Control. 2002;**75**(18):1472-1485. DOI: 10.1109/87.865858

[18] Haykin S. Kalman Filtering and Neural Networks, Adaptive and Learning Systems for Signal Processing, Communications, and Control. New York, USA: Wiley–Interscience, John Wiley & Sons, Inc; 2001

[19] Xu J-X, Liu C, Hang C. Combined adaptive and fuzzy control using multiple models. In: Proceedings of Third IEEE International Conference on Fuzzy Systems, Orlando, FL. 1994

[20] Hunt K, Sbarbaro D, Zbikowki R, Gawthrop P. Neural networks for control system: A survey. IEEE Transactions on Neural Networks. 1992;**28**:1083-1112

[21] Simani S, Farsoni S, Castaldi P. Fault diagnosis of a wind turbine benchmark via identified fuzzy models. IEEE Transactions on Industrial Electronics. 2015;**62**(6):3775-3782. Invited paper for the special issue “Real–time fault diagnosis and fault tolerant control”. DOI: 10.1109/TIE.2014.2364548

[22] Ding SX. Model–Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools. 1st ed. Berlin Heidelberg: Springer; 2008. ISBN: 978–3540763031