Int. J. Appl. Math. Comput. Sci., 2018, Vol. 28, No. 2, 247–268
DOI: 10.2478/amcs-2018-0018

DATA–DRIVEN TECHNIQUES FOR THE FAULT DIAGNOSIS OF A WIND TURBINE BENCHMARK

SILVIO SIMANI [a,*], SAVERIO FARSONI [a], PAOLO CASTALDI [b]

[a] Department of Engineering
University of Ferrara, Via Saragat 1/E, 44124 Ferrara, Italy
e-mail: {silvio.simani,saverio.farsoni}@unife.it

[b] Department of Electronics, Computer Science and Systems
University of Bologna, Via Fontanelle 40, 47100 Forlì, Italy
e-mail: paolo.castaldi@unibo.it

This paper deals with the fault diagnosis of wind turbines and investigates viable solutions to the problem of earlier fault detection and isolation. The design of the fault indicator, i.e., the fault estimate, involves data-driven approaches, as they can represent effective tools for coping with poor analytical knowledge of the system dynamics, together with noise and disturbances. In particular, the proposed data-driven solutions rely on fuzzy systems and neural networks that are used to describe the strongly nonlinear relationships between measurement and faults. The chosen architectures rely on nonlinear autoregressive models with exogenous input, as they can represent the dynamic evolution of the system along time. The developed fault diagnosis schemes are tested by means of a high-fidelity benchmark model that simulates the normal and the faulty behaviour of a wind turbine. The achieved performances are also compared with those of other model-based strategies from the related literature. Finally, a Monte-Carlo analysis validates the robustness and the reliability of the proposed solutions against typical parameter uncertainties and disturbances.

Keywords: fault diagnosis, analytical redundancy, fuzzy systems, neural networks, residual generators, fault estimation, wind turbine benchmark.

1. Introduction

The increased level of wind-generated energy in power grids worldwide raises the levels of reliability and sustainability required of wind turbines. Wind farms should have the capability to generate the desired value of electrical power continuously, depending on the actual wind speed level and on the grid's demand.

As a consequence, the possible faults affecting the system have to be properly identified and treated before they endanger the correct functioning of the turbines or become critical faults. Megawatt-class wind turbines are extremely expensive systems. Therefore, their availability and reliability must be high, in order to assure the maximisation of the generated power while minimising the operation and maintenance (O & M) services. Alongside the fixed costs of the produced energy, mainly due to the installation and the foundation of the wind turbine, the O & M costs could increase the total energy cost up to about 30%, particularly considering the offshore installation (Odgaard, 2012).

These considerations motivate the introduction of a fault diagnosis system coupled with fault tolerant controllers. Currently, most of the turbines feature a simply conservative approach against faults that consists in the shutdown of the system to wait for maintenance service. Hence, effective strategies coping with faults have to be studied and developed for improving the turbine performance, particularly in faulty working conditions. Their benefits would concern the prevention of critical failures that jeopardise wind turbine components, thus avoiding an unplanned replacement of functional parts, as well as reduction in the O & M costs and an increase in the energy production. The advent of computerised control, communication networks and information techniques brings interesting challenges concerning the development of novel real-time monitoring

*Corresponding author

and fault tolerant control design strategies for industrial processes.

Indeed, in recent years, many contributions have been proposed related to the topics of fault diagnosis of wind turbines (see, e.g., Chen *et al*., 2011; Gong and Qiao, 2013). Some of them highlight the difficulties to achieve the diagnosis of particular faults, e.g., those affecting the drive train, at the wind turbine level. However, these faults are better dealt with at the wind farm level, when the wind turbine is considered in comparison to another wind turbine of the wind farm (Odgaard and Stoustrup, 2013). Moreover, fault tolerant control of wind turbines has been investigated, e.g., by Odgaard and Stoustrup (2015) or Parker *et al*. (2011), and international competitions on these issues have been organised (Odgaard and Stoustrup, 2012; Odgaaard and Shafiei, 2015).

Hence, the fault diagnosis in connection with the *sustainable* control of dynamic processes has been proven to be a challenging task (see, e.g., Hassanabadi *et al*., 2016; Byrski and Byrski, 2016; Xu *et al*., 2017), especially for energy conversion systems, such as wind turbines, and has thus motivated the research activities carried out through this paper.

In recent years, the increasing demand for energy generation from renewable sources has led to close attention on wind turbines. Indeed, they represent very complex systems which require reliability, availability, maintainability, safety and, above all, efficiency on the generation of electrical power. Thus, new research challenges arise, in particular in the context of modelling and control. Advanced sustainable control systems can provide the optimisation of energy conversion and guarantee the desired performances even in the presence of possible anomalous working conditions caused by unexpected faults and disturbances.

This work deals with the fault diagnosis for wind turbine systems, and it proposes the application of viable and reliable solutions to the problem of earlier fault detection and isolation (FDI). Further fault tolerant controllers, which are not considered in this work, can be based on the fault diagnosis module developed in this paper, which provides the on-line information on the faulty or fault-free status of the system, so that the controller action can be compensated. The design of the *fault estimators* (or reconstructors) that are used for the FDI task involves data-driven approaches, as they offer an effective tool for coping with a poor analytical knowledge of the system dynamics, together with noise and disturbances.

The first data-driven proposed solution relies on fuzzy Takagi–Sugeno models that are derived from a clustering c-means algorithm, followed by an identification procedure solving the noise-rejection problem. Then, a second solution makes use of neural networks to describe the strongly nonlinear relationships between measurement and faults. The chosen network architecture belongs to the nonlinear autoregressive with exogenous (NARX) input topology, as it can represent a dynamic evolution of the system along time. The training of the neural network fault estimators exploits the classical back-propagation Levenberg–Marquardt algorithm that processes a set of acquired target data.

A purely nonlinear model-based scheme for fault tolerant control was also proposed by the same authors, which is based on differential algebraic tools relying on a nonlinear geometric approach. Already suggested in the aerospace framework, it was extended by the same authors to the active fault tolerant control for the same wind turbine benchmark and a wind farm (Simani and Castaldi, 2014). Other important contributions by the same authors are, e.g., those by Simani and Turhan (2017), Castaldi *et al*. (2017) or Simani and Castaldi (2018).

The developed fault diagnosis schemes are tested by means of a high-fidelity benchmark model that simulates the normal and the faulty behaviour of a single wind turbine. The achieved performances are compared with those of other solutions from the related literature. Moreover, a Monte-Carlo (MC) analysis validates the robustness of the proposed systems against the typical parameter uncertainties and disturbances. Finally, the hardware-in-the-loop test is carried out in order to assess the performance in a more realistic real-time framework. The effectiveness shown by the achieved results suggests further investigations on the industrial application of the proposed systems.

The work is organised as follows. Section 2 recalls the wind turbine benchmark simulator. Section 3 describes the fault diagnosis scheme relying on fuzzy systems and neural network structures. The achieved results are reported in Section 4. Comparisons with various FDI strategies are also reported. Finally, Section 5 concludes the paper by summarising the main achievements of the work, and providing some suggestions for further research topics.

2. Wind turbine benchmark

The wind turbine benchmark model adopted in this study was proposed by Odgaard *et al*. (2013). It provides a simulator of a three-blade horizontal-axis variable-speed pitch-controlled turbine with a full converter generator. The overall model consists of four interconnected submodels, namely the wind, the turbine, the measurement and the controller modules. The turbine model is composed of three further submodels: the blade and the pitch, the drive train and the generator. The relations among these submodels are depicted in the block diagram of Fig. 1. Moreover, several fault scenarios can be simulated (Odgaard *et al*., 2013).

The following sections address the description of the

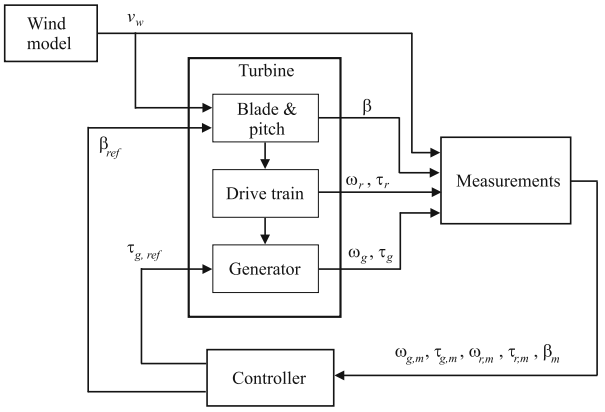

Fig. 1. Block diagram of wind turbine benchmark subsystems.

four interconnected submodels.

2.1. Wind model. Wind is considered as a stochastic process driven by a mean speed sequence coming from real on-the-field acquisitions. In addition, the wind model includes the effects of wind shear and tower shadow (Dolan and Lehn, 2006). Therefore, the actual wind speed v_w is the sum of four terms:

$$v_w(t) = v_m(t) + v_s(t) + v_{ws}(t) + w_{ts}(s), \qquad (1)$$

where v_m is the mean speed, v_s is the stochastic component, described as a Gaussian white noise, v_{ws} takes into account the wind shear effect due to the distance from the earth surface and w_{ts} represents the tower shadow effect, i.e., the wind speed reduction when a blade passes in front of the tower. Wind shear variations and the tower shadow effect are not measured by the anemometer, because it is supposed to be placed on top of the nacelle.

2.2. Turbine model. As already remarked, the turbine system can be further subdivided into three submodels focused on the power transmission. Firstly, the blade and pitch models describe how the blades capture wind energy. This phenomenon is based on the aerodynamic law

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

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

$$\frac{\beta(s)}{\beta_{\text{ref}}(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \qquad (3)$$

where β_{ref} is the pitch angle target value elaborated by the turbine controller, ζ and ω_n are the transfer function parameters.

The second submodel represents the drive train, i.e., the power flow through the gear-box from the rotor toward the generator, whose dynamics can be described as follows:

$$\begin{aligned} J_r\dot{\omega}_r &= \tau_r - K_{dt}\theta_\Delta - (B_{dt} + B_r)\omega_r \\ &\quad + \frac{B_{dt}}{N_g}\omega_g, \\ J_g\dot{\omega}_g &= \frac{\eta_{dt}K_{dt}}{N_g}\theta_\Delta + \frac{\eta_{dt}B_{dt}}{N_g}\omega_r \\ &\quad - \Big(\frac{\eta_{dt}B_{dt}}{N_g^2} + B_g\Big)\omega_g - \tau_g, \\ \dot{\theta}_\Delta &= \omega_r - \frac{\omega_g}{N_g}, \end{aligned} \qquad (4)$$

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 third submodel provides the generator and the converter dynamics by means of the following first-order transfer function:

$$\frac{\tau_g(s)}{\tau_{g,\text{ref}}(s)} = \frac{\alpha_g}{s + \alpha_g}, \qquad (5)$$

where $\tau_{g,\text{ref}}$ is the target torque coming from the controller and α_g is the transfer function parameter.

The final generated power P_g can be computed as the product of the generator torque and speed, decreased by the efficiency coefficient η_g

$$P_g = \eta_g\,\omega_g\,\tau_g. \qquad (6)$$

The signals that are assumed to be acquired from the wind turbine simulator for monitoring are processed through the measurement model whose objective is to simulate real sensor and actuator behaviours. Therefore, the measured signals are the sum of their actual value and white Gaussian noise terms. The list of the available sensors is summarised in Table 1. Note that some of these measured signals are redundant, as for the case of the sensors measuring the pitch angles of the blades.

2.3. Controller. The model of the wind turbine compensator implemented in the benchmark model

Table 1. List of available sensors in the wind turbine simulator.

Measurements	Description
$\beta_{1,m1}$	Blade 1 pitch angle
$\beta_{1,m2}$	Blade 1 pitch angle
$\beta_{2,m1}$	Blade 2 pitch angle
$\beta_{2,m2}$	Blade 2 pitch angle
$\beta_{3,m1}$	Blade 3 pitch angle
$\beta_{3,m2}$	Blade 3 pitch angle
$\omega_{r,m1}$	Rotor shaft speed
$\omega_{r,m2}$	Rotor shaft speed
$\omega_{g,m1}$	Generator shaft speed
$\omega_{g,m2}$	Generator shaft speed
$\tau_{g,m}$	Generator shaft torque
$P_{g,m}$	Generated power
$v_{w,m}$	Wind speed at hub height
$\tau_{r,m}$	Aerodynamic rotor torque

describes the behaviour of the baseline wind turbine controller that regulates the generated power on the basis of the actual wind speed (Odgaard *et al*., 2013; Odgaard and Stoustrup, 2015). In more detail, the reference signal for generated power as a function of the wind speed is depicted in Fig. 2, where four different regions corresponding to the different operating points of a real wind turbine can be identified:

- $v_w < v_{\text{in}}$: the turbine is in idle state because of the low value of wind speed;
- $v_{\text{in}} < v_w < v_{\text{nom}}$, *partial load* region: for a given wind speed value the controller acts on the generator torque so that the maximum available power is tracked; pitch angles are kept to their optimal value ($\beta = 0$) in order to capture the maximum aerodynamic power from the wind;
- $v_{\text{nom}} < v_w < v_{\text{out}}$, *full load* region: after reaching the nominal generated power, corresponding to the nominal wind speed, the controller regulates the blade pitch angles to limit the generated power to its nominal value despite of the increasing of wind speed;
- $v_w > v_{\text{out}}$: turbine is turned off to prevent critical damages.

2.4. Fault scenarios. Three different groups of typical faults that affect a wind turbine can be simulated, i.e., sensor, actuator and system faults (Odgaard *et al*., 2013; Odgaard and Stoustrup, 2015).

Sensor faults are modelled as additive signals on the affected measurements. For example, a faulty pitch angle sensor provides wrong measurements of the blade orientation. Thus, if not handled, the controller cannot fully track the power reference signal.

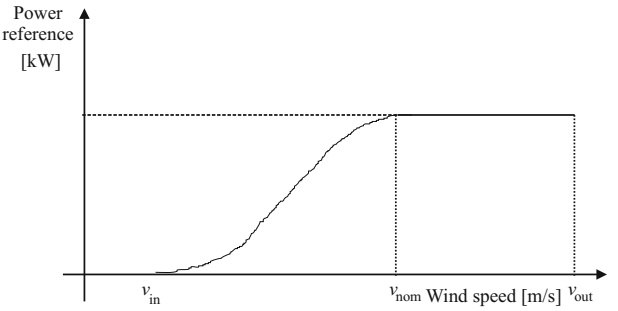

Fig. 2. Controller reference curve. The default values are cut-in speed $v_{\text{in}} = 3$ m/s, nominal speed $v_{\text{nom}} = 12.5$ m/s, cut-out speed $v_{\text{out}} = 25$ m/s, nominal power $P_{\text{nom}} = 4.8$ MW.

Actuator faults involve modifications of the pitch angle or generator torque transfer functions of Eqns. (3) and (5) in the form of changed dynamics. They represent, e.g., a pressure drop in the hydraulic pitch actuator or an electronic break-down in the converter equipment.

Finally, the system fault can affect the drive-train of the turbine. It is modelled as a slow time variation in the friction coefficient. This corresponds to the effect of wear and tear along time. These nine fault cases included in the wind turbine benchmark model are summarised in Table 2, together with the affected measured signals.

Table 2. Fault scenario of the wind turbine simulator.

Fault case	Type	Description
1	Sensor	Faulty $\beta_{1,m1}$ sensor
2	Sensor	Faulty $\beta_{2,m2}$ sensor
3	Sensor	Faulty $\beta_{3,m1}$ sensor
4	Sensor	Faulty $\omega_{r,m1}$ sensor
5	Sensor	Faulty $\omega_{r,m2}$ and $\omega_{g,m2}$ sensors
6	Actuator	Air content in oil, faulty response of Blade 2 pitch system
7	Actuator	Low pressure, faulty response of Blade 3 pitch system
8	Actuator	Fault on converter, faulty response of the generator torque
9	System	Drive train wear and tear, slow friction variations

With these assumptions, the complete model of the system under analysis (Odgaard and Stoustrup, 2015) can be represented by means of a non-linear continuous-time function $\mathbf{f}_{wt}$ that describes the evolution of the turbine state vector $\mathbf{x}_{wt}$ excited by the input vector $\mathbf{u}$:

$$\begin{cases} \dot{\mathbf{x}}_{wt}(t) = \mathbf{f}_{wt}(\mathbf{x}_{wt}, \mathbf{u}(t)), \\ \mathbf{y}(t) = \mathbf{x}_{wt}(t), \end{cases} \tag{7}$$

where the state of the system is assumed to be equal to the monitored system output, i.e., the rotor speed, the

generator speed and the generated power:

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

On the other hand, the input vector

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

contains the measurements of the pitch angles from the three sensor couples as well as the measured torque. These vectors are sampled for obtaining N input-output data $\mathbf{u}(k)$, $\mathbf{y}(k)$ with $k = 1, \ldots, N$, in order to implement the data-driven estimators at the sampling time T.

3. Fault diagnosis techniques and design

This work exploits two data-driven approaches based on fuzzy systems and neural networks to implement the fault diagnosis block. In this section, after a brief introduction on the general structure of a fault diagnosis system, recalled in Section 3.1, the properties and the structure of fuzzy systems and neural networks are addressed in Sections 3.2 and 3.4, respectively. In particular, their architectures of the NARX systems are reported, since they represent, in combination with proper training algorithms, the exploited solutions for the implementation of the fault estimators used as residual signals for fault detection and isolation.

3.1. Fault diagnosis scheme. In the following the monitored system, i.e., the wind turbine under diagnosis, is assumed to be affected by additive faults on the input (actuator) and output (sensor) measurements, as well as measurement noise, as represented in Fig. 3, in the form

$$\begin{cases} \mathbf{u}(k) = \mathbf{u}^*(k) + \tilde{\mathbf{u}}(k) + \mathbf{f}_u(k), \\ \mathbf{y}(k) = \mathbf{y}^*(k) + \tilde{\mathbf{y}}(k) + \mathbf{f}_y(k), \end{cases} \tag{8}$$

where $\mathbf{u}^*(k)$ and $\mathbf{y}^*(k)$ are the actual unmeasurable variables, $\mathbf{u}(k)$ and $\mathbf{y}(k)$ represent the acquired sensor measurements affected by the measurement noise signals $\tilde{\mathbf{u}}(k)$ and $\tilde{\mathbf{y}}(k)$. According to (8), also the faults $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$ have additive effects, and they are different from zero only in the presence of faults. In general, the vector $\mathbf{u}(k)$ has r components, *i.e.* the number of the process inputs, while $\mathbf{y}(k)$ has m elements, i.e., the number of process outputs.

Figure 3 shows the general scheme with faults $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$ affecting the system under diagnosis, i.e., the wind turbine simulator, as *equivalent* additive signals on the input and output measured signals. The same effect is assumed for the measurement noise $\tilde{\mathbf{u}}(k)$ and $\tilde{\mathbf{y}}(k)$, as

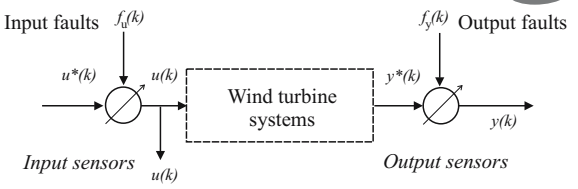

Fig. 3. Measurement process in the system under diagnosis.

described in Section 2.2. It is worth noting that, according to (8), as shown also in Fig. 3, it is assumed that the *equivalent* effect of the considered faults is additive on both the input and output measurements. This represent the key point for the fault sensitivity analysis reported in Section 3.6 and the fault estimator design addressed in the sequel. In general, the number of the process inputs is r, while the number of outputs is m.

Note that the equivalent and additive representation proposed in Eqn. (8) represents one of the key points of the paper. Moreover, some of the considered faults considered in Table 2 are not modelled in an additive way. However, as described in this section, this work assumes that the effects of the considered fault cases are described by considering the equivalent effects of additive signals $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$ in (8) on the input and output measurements. This assumption is also consistent with the modelling framework considered in this paper that relies on the Frisch scheme (Beghelli *et al.*, 1990) usually exploited in connection with the identification of errors-in-variables (EIV) models (Fantuzzi *et al.*, 2002), as remarked in Section 3.2.

Among the different approaches to generate the residual signals recalled in Section 1, the solution adopted in this work exploits both the fuzzy system and neural network models, which provide an on-line estimate $\hat{\mathbf{f}}(k)$ of the faulty signals $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$. Hence, as shown in Fig. 4, the residuals $\mathbf{r}$ are assumed to be equal to the estimated fault signals, $\hat{\mathbf{f}}(k)$, which are generated by the generic fault estimator, as

$$\mathbf{r}(k) = \hat{\mathbf{f}}(k), \tag{9}$$

$\hat{\mathbf{f}}(k)$ being the generic fault vector, i.e.,

$$\hat{\mathbf{f}}(k) = \left\{ \hat{f}_1(k), \ldots, \hat{f}_{r+m}(k) \right\}.$$

Therefore, the generic fault estimate $\hat{f}_i(k)$ can be equal to one of the i-th component of the fault vectors $\mathbf{f}_u(k)$ or $\mathbf{f}_y(k)$ in (8), with $i = 1, \ldots, r+m$. The general residual generation scheme that exploits the fault estimation as the residual generator is sketched in Fig. 4. Note that this approach is able to provide both the fault detection and isolation tasks, as addressed in the following.

Figure 4 sketches the residual generation scheme that is achieved via the fault estimator system, by using

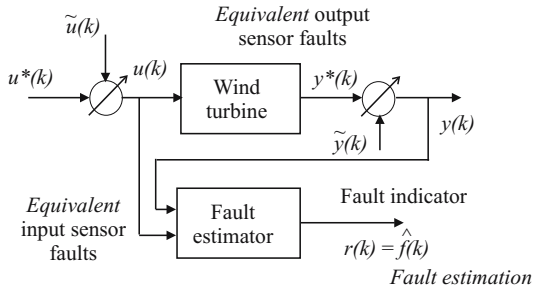

Fig. 4. General residual generation scheme relying on the fault estimation strategy.

the acquired input and output measurements $\mathbf{u}(k)$ and $\mathbf{y}(k)$. The fault diagnosis process requires, as the first step, the fault detection task. As the residual is equal to the estimated fault signal, it is easily performed here by using a proper thresholding logic directly operating on the residuals, without requiring their elaboration with a proper evaluation function, as addressed, e.g., by Chen and Patton (1999).

Therefore, the occurrence of the i-th fault can be simply detected via the following threshold logic applied to the i-th residual $r_i(k)$:

$$\begin{cases} \bar{r}_i - \delta\sigma_{r_i} \le r_i \le \bar{r}_i + \delta\sigma_i, & \text{fault-free case,} \\ r_i < \bar{r}_i - \delta\sigma_{r_i} \text{ or } r_i > \bar{r}_i + \delta\sigma_{r_i}, & \text{faulty case,} \end{cases} \tag{10}$$

where the i-th component $r_i(k)$ of the residual vector $\mathbf{r}(k)$ is considered a random variable whose unknown mean $\bar{r}_i$ and variance $\sigma^2_{r_i}$ can be estimated in a fault-free condition, after the acquisition of N samples, as

$$\begin{cases} \bar{r}_i = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} r_i(k), \\ \sigma^2_{r_i} = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} (r_i(k) - \bar{r}_i)^2. \end{cases} \tag{11}$$

The tolerance parameter $\delta \ge 2$ has to be properly tuned in order to separate the fault-free and faulty conditions. The value δ determines the trade-off between the false alarm rate and the fault detection probability. A common choice of δ can rely on the three-sigma rule, otherwise extensive simulations can be performed to optimise δ.

After fault detection, the fault isolation task is easily accomplished by means of a bank of estimators. As described by (8), the faults are considered as equivalent signals that affect the input measurements, i.e., $\mathbf{f}_u$, or the output measurements, i.e., $\mathbf{f}_y$.

Under this assumption, by following the scheme of the generalised estimator configuration of Fig. 5, in order to uniquely isolate one of the input or output faults, by considering that multiple faults cannot occur, a bank of multi-input single-output (MISO) fault estimators is used. In general, the number of these estimators is equal to the number of faults that have to be diagnosed, i.e., equal to the number of input and output measurements, $r + m$. Therefore, in general, the i-th fault estimator that reconstructs the fault $\hat{f}(k) = r_i(k)$ is driven by the components of the input and output signals $\mathbf{u}(k)$ and $\mathbf{y}(k)$ that are sensitive to the specific fault $f_i(t)$. Therefore, it should be clear that the design of these fault estimators is enhanced by the so-called failure mode & effect analysis (FMEA) described in Section 3.6. For each fault case, the failure modes and their resulting effects on the rest of the system are analyzed and, in particular, the most sensitive input and output measurements to that specific fault situation are identified. 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 and the faults, as represented by the estimator bank of Fig. 5.

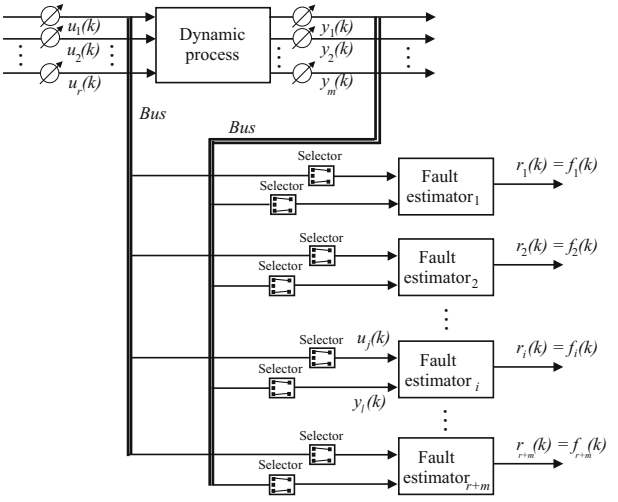

Fig. 5. General estimator scheme for the reconstruction of the *equivalent* input or output fault.

Figure 5 shows this generalised fault estimator scheme, where the fault estimators are driven only by the input-output signals selected via the FMEA tool, so that the relative residual $r_i(k) = \hat{f}_i(k)$ is insensitive only to the fault affecting those inputs and outputs defined by the selector blocks. It is worth noting that multiple faults occurring at the same time cannot be correctly isolated, using this configuration.

As for the structure of the residual generators, in general, the number of these fault estimation filters is equal to the number of faults that have to be diagnosed, i.e., $m + r$. In fact, the residual generator functions are MISO discrete-time models fed by the measurements selected by the fault sensitivity analysis. Moreover, their outputs provide the estimated faults, which are equal to the number of input and output measurements, i.e., $m+r$. Moreover, the number of inputs that feed the residual

generators is equal to the number of input and output measurements selected from the fault sensitivity analysis that will be described in Section 3.6. Moreover, the number of parameters of each MISO residual generator (used for fault estimation) in general is equal to the number of inputs + 1 multiplied by the number of delays (n) selected for each residual generator input. On the other hand, the complexity of the neural networks used for residual generation depends on the number of neurons for each layer of the neural network. Moreover, also in this case, the number of neural networks is equal to the number of faults that have to be diagnosed, i.e., $m + r$.

As already remarked, the FMEA tool (Stamatis, 2003), which has to be executed before the design of the fault estimators, suggests how to select the input-output configuration for the fault estimator blocks. This input-output selection procedure, which is briefly remarked here, will be addressed in more detail in Section 3.6. Then, the design of the fuzzy or neural network models can be performed, as recalled in Sections 3.2 and 3.4, respectively. Finally, the threshold test logic of (10) allows the achievement of the fault diagnosis tasks.

Finally, with reference to the FDI approach considered in this work, it is worth noting that model uncertainty is not only related to the effects of noise and disturbance, but also to the so-called model-mismatches. The use of fixed thresholds as proposed in this paper may lead to conservative results. However, as will be presented in Section 4, the robustness of the proposed solutions have been analysed both in simulations and by means of hardware-in-the-loop (HIL) tests. Moreover, the fixed thresholds considered in the logic of (10) have been settled by means of a procedure using again the Monte-Carlo tool, which will lead to the optimisation of the performance indices discussed in Section 4.4. This procedure was already proposed by the same authors with application to the aerospace framework. It was shown that this optimisation strategy exploiting the Monte-Carlo tool itself was able to outperform more conservative results achieved for example via H_∞ robust methodologies (Patton *et al.*, 2008; 2010). Moreover, an example with the comparison of variable thresholds will be shown in Section 4. However, as was remarked, e.g., by Chen and Patton (1999), when the residual signals are the estimated faults themselves, the threshold settlement issue does not represent the point of the discussion, since the fault sensitivity is maximised via the FMEA procedure proposed in the paper, which inherently takes into account modelling and measurement errors.

3.2. Fuzzy system modelling. This section describes the design of the dynamic estimators by means of the Takagi–Sugeno (TS) prototypes. Indeed, the unknown relationships between noisy measurements and faults are provided by fuzzy models, which consist of a number of rules connecting the measured signals acquired from the system under investigation to its faults, on the basis of knowledge of the system dynamics, in the form of IF-THEN relations, processed by a fuzzy inference system (FIS) (Babuška, 1998).

3.3. Takagi–Sugeno fuzzy prototypes. Approximation of nonlinear MISO systems (but also extension to MIMO systems can be considered) can be achieved by the Takagi–Sugeno (TS) fuzzy reasoning, as reported, e.g., by Fantuzzi and Rovatti (1996) or Rovatti (1996). According to the TS modelling approach (Takagi and Sugeno, 1985), the consequents become crisp functions of the input, while the antecedents remain fuzzy propositions. Therefore the fuzzy rule takes the form

$$\begin{aligned} R_i: \quad & \text{IF} \quad (\text{fuzzy combination of inputs}) \\ & \text{THEN output} = g_i(\text{inputs}), \end{aligned} \tag{12}$$

where i indicates the number of rules. The antecedent does not differ from the Mamdani rules, with a combined membership function $\lambda_i(\mathbf{x})$ that takes into account the logical connectives expressed by linguistic propositions. The rule consequent function $g_i(\cdot)$ has a defined structure: it is an instance of a parametrised function in the affine form

$$g_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} + b_i, \tag{13}$$

where $\mathbf{a}_i$ is a parameter vector and b_i is a scalar offset, while $g_i(\mathbf{x})$ is the output of the i-th rule. The number of rules is supposed equal to the number of clusters n_C used for partitioning the data into regions where local linear relations can be assumed (Babuška, 1998). Furthermore, the antecedent of each rule defines the degree of fulfilment for the corresponding consequent model, so that the rule global model can be seen as a fuzzy composition of linear local models.

Thus, the TS inference takes the form of the simple algebraic expression

$$\hat{f} = \frac{\sum_{i=1}^{n_C} \lambda_i(\mathbf{x})\, g_i(\mathbf{x})}{\sum_{i=1}^{n_C} \lambda_i(\mathbf{x})} \tag{14}$$

The estimated fault $\hat{f}$ is the weighted average of affine functions of the input-output measurements, where the weights are the combined degree of fulfilment of the system input.

It is worth noting that the nonlinear system under investigation described in Section 2 has a dynamic behaviour. Therefore, the considered input vector $\mathbf{x}$ of the model can contain the current as well as previous samples of the system input and output signals.

Indeed, in order to introduce the time dependence into the model (12), the consequents are considered

as discrete-time linear autoregressive models with exogenous input (ARX) of order o, in which the regressor vector takes the form

$$\mathbf{x}(k) = \Big[\ldots, y_i(k-1), \ldots, y_i(k-o), \ldots \\ \ldots, u_j(k), \ldots, u_j(k-o), \ldots\Big]^T \quad (15)$$

where $u_i(\cdot)$ and $y_j(\cdot)$ are the components of the actual system input and output vectors $\mathbf{u}(k)$ and $\mathbf{y}(k)$ selected via the FMEA procedure of Section 3.6, and k is the time step, with $k = 1, 2, \ldots, N$. The affine parameters of (13) can be grouped into

$$\mathbf{a}_i = \Big[\alpha_1^{(i)}, \ldots, \alpha_o^{(i)}, \delta_1^{(i)}, \ldots, \delta_o^{(i)}\Big]^T, \quad (16)$$

where the coefficients $a^{(i)}$ are associated with the output samples, and the $\delta^{(i)}$ are associated to the input ones.

Note that the representation (14) does not contain algebraic loops in the system, even if the input u is a premise variable. This is prevented in practice by introducing at least a unit delay in the time-discrete model of (13). This is achieved during the design of the identification experiment, and the possible presence of algebraic loops is thus avoided by a proper selection of the time dependence in the dynamic model (14) and (15), where the consequents are considered as discrete-time linear autoregressive models with exogenous input (ARX) of order o, in which the regressor vector takes the form of the relation (15). Therefore, a proper estimation of the delays of the samples of the input-output signals allows us to overcome this problem. An example of an identified residual generator highlighting this feature will be provided in Section 4.

An effective approach to the design of an FIS as an approximator of a complex nonlinear system begins with the partitioning of the available data into subsets characterised by a simpler (affine) behaviour. A cluster can be defined as a group 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.

Indeed, fuzzy clustering allows an item to belong to several cluster simultaneously, with different degrees of fulfilment, whereas the classical various crisp clustering relies on mutually exclusive subsets. Various clustering methods have been proposed in the literature; see, e.g., the review by Jain and Dubes (1988), or the more recent works by Jun *et al*. (2011) as well as Graaff and Engelbrecht (2012).

Typically, the available data consist of noisy measurements acquired from the system. They are grouped into the data matrix $\mathbf{Z}$ whose columns are the vectors z containing the measurements of a single observation of the system under analysis:

$$\mathbf{Z} = \begin{bmatrix} z_{11} & \ldots & z_{1N} \\ \vdots & \ddots & \vdots \\ z_{n1} & \ldots & z_{nN} \end{bmatrix}, \quad (17)$$

where n is the data dimension, N is the number of available observations.

Most fuzzy clustering algorithms are based on the optimisation of the c-means goal function $J(\mathbf{Z}, \mathbf{U}, \mathbf{V})$ performed as follows:

- The data matrix $\mathbf{Z}$ is defined.
- The so-called fuzzy partition matrix $\mathbf{U} = [\mu_{ik}]$ is defined, which contains the values of the membership function μ_{ik} for the couple i-th measurement/k-th cluster.
- The vector $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_{n_C}]$ containing the cluster prototypes is defined. It has to be determined since it consists of the centres from which the distance of each measurement can be calculated.

The widespread c-means goal function adopted in this work was formulated by Bezdek (1981) in the form

$$J(\mathbf{Z}, \mathbf{U}, \mathbf{V}) = \sum_{i=1}^{n_C} \sum_{k=1}^{N} (\mu_{ik})^m \, D_{ik\mathbf{A}}^2, \quad (18)$$

$m > 1$ being the weighting exponent, and

$$D_{ik\,\mathbf{A}}^2 = \|\mathbf{z}_k - \mathbf{v}_i\|_{\mathbf{A}}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{z}_k - \mathbf{v}_i) \quad (19)$$

is a squared inner product distance norm, with $i = 1, \ldots, n_C$ and $k = 1, \ldots, N$. The matrix $\mathbf{A}$ determines the cluster shape.

The minimisation algorithm exploits a series of Picard iterations consisting in the updating of the cluster prototypes and of the partition matrix, until the stopping criterion is met, as addressed, e.g., by Babuška (1998).

A key aspect concerns the determination of the optimal number of clusters n_C, as the clustering algorithm assumes that a proper number of clusters n_C has been fixed, regardless of whether or not they are really present in the data. Once the partition matrix has been estimated, the antecedent degrees of fulfilment μ_{ik} are easily derived by interpolation or curve fitting methods (Babuška, 1998).

Then, the design of the FIS assumes the form of a system identification problem from noisy data, as it requires the estimation of the consequent parameters $\mathbf{a}_i$ and b_i of (13) using the input-output data. The identification scheme adopted in this work was proposed by the same authors (Rovatti *et al*., 2000) and successfully exploited for the approximation of nonlinear functions

through piecewise affine models (Fantuzzi *et al*., 2002). 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 so-called Frisch scheme (Beghelli *et al*., 1990) that is usually exploited in connection with the identification of errors-in-variables models (Fantuzzi *et al*., 2002).

In fact, by considering a discrete-time MISO system, as already described by the relations (8), the noise is supposed to affect the input $\mathbf{u}$ as well as the output y measurements in the form of the additive signals $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{y}}$ on the noise-free unmeasurable quantities $\mathbf{u}^*$ and $\mathbf{y}^*$ in the form of

$$\begin{cases} \mathbf{u}(k) = \mathbf{u}^*(k) + \tilde{\mathbf{u}}(k), \\ \mathbf{y}(k) = \mathbf{y}^*(k) + \tilde{\mathbf{y}}(k). \end{cases} \tag{20}$$

Thus, considering the i-th TS consequent of the type of (14) and the associated dynamic local ARX model of order o with the regressors grouped into the vector $\mathbf{x}$ as in (15), the acquisition of N_i noisy measurement of input and output samples permits the construction of the i-th data matrix

$$\mathbf{X}^{(i)} = \begin{bmatrix} f_j(k) & \mathbf{x}^T(k) & 1 \\ f_j(k+1) & \mathbf{x}^T(k+1) & 1 \\ \vdots & \vdots & \vdots \\ f_j(k+N_i-1) & \mathbf{x}^T(k+N_i-1) & 1 \end{bmatrix}. \tag{21}$$

Note that the matrix $\mathbf{X}^{(i)}$ contains the delayed samples of the j-th fault signal to be reconstructed, which is related with the choice of the input-output measurements in the vector $\mathbf{x}(k)$ depending on the FMEA selection procedure. The i-th covariance matrix from the acquired data can be computed as

$$\mathbf{\Sigma}^{(i)} = {\mathbf{X}^{(i)}}^T \mathbf{X}^{(i)} \geq 0, \tag{22}$$

which is a positive-definite matrix consisting of two terms

$$\mathbf{\Sigma}^{(i)} = \mathbf{\Sigma}^{(i)^*} + \bar{\tilde{\mathbf{\Sigma}}}^{(i)}, \tag{23}$$

where $\mathbf{\Sigma}^{(i)^*}$ refers to the noise-free signals, while $\bar{\tilde{\mathbf{\Sigma}}}^{(i)}$ is the noise covariance matrix, which depends on the unknown noise variances $\bar{\tilde{\sigma}}_{\mathbf{f}}$,$\bar{\tilde{\sigma}}_x$ through

$$\bar{\tilde{\mathbf{\Sigma}}}^{(i)} = \mathbf{diag}\left[\bar{\tilde{\sigma}}_f \mathbf{I},\, \bar{\tilde{\sigma}}_{\mathbf{x}} \mathbf{I},\, 0\right]. \tag{24}$$

The solution of the identification problem mentioned above requires estimation of $\bar{\tilde{\sigma}}_f$ and $\bar{\tilde{\sigma}}_x$, which can be performed by solving

$$\mathbf{\Sigma}^{(i)^*} = \mathbf{\Sigma}^{(i)} - \tilde{\mathbf{\Sigma}}^{(i)} \tag{25}$$

with

$$\tilde{\mathbf{\Sigma}}^{(i)} = \mathrm{diag}\left[\tilde{\sigma}_f \mathbf{I},\, \tilde{\sigma}_{\mathbf{x}} \mathbf{I},\, 0\right] \tag{26}$$

in the variables $\tilde{\sigma}_{\mathbf{x}}$ and $\tilde{\sigma}_f$. Note that these terms represent the uncertainty affecting the input-output measurements $\mathbf{u}(k)$ and $\mathbf{y}(k)$ and the fault signal $f_j(k)$.

In case all the assumption regarding the Frisch scheme (Rovatti *et al*., 2000) are satisfied, there exists one common point belonging to all the surfaces $\mathbf{\Gamma}^{(i)} = 0$ determined as the root locus of (25), which represents the actual noise variance values $(\bar{\tilde{\sigma}}_{\mathbf{x}},\, \bar{\tilde{\sigma}}_f)$. However, in real cases, the Frisch assumptions are commonly violated, so that a unique solution cannot be obtained. In these situations the identification aims at finding the nearest point of all the surfaces.

After the computation of the variances, the covariance noise matrix can be built as in (24), and the linear parameters in each cluster (therefore in each TS consequent) can be finally determined as the consistent solution of the following expression (Rovatti *et al*., 2000):

$$\left(\mathbf{\Sigma}^{(i)} - \bar{\tilde{\mathbf{\Sigma}}}^{(i)}\right) \mathbf{a}_i = 0, \tag{27}$$

$i = 1, \ldots, n_C$.

Note finally that due to both the structure of the fuzzy fault estimators and the fault models, their effect does not vanish in the fuzzy ARX filters, since no integral action is present. In general, the fuzzy ARX filters perform a processing of the input and output measurements in order to reconstruct the fault functions. Moreover, their parameters are estimated off-line and they are not adapted during the on-line fault estimation task. However, due to the ARX structure of the fuzzy models, delays can be present, as analysed in Section 4, even if they satisfy the application requirements.

3.4. Neural network modelling. Alongside the fuzzy models, a different data-driven approach, based on neural networks, has been proposed in order to implement the fault diagnosis block. In this section, after a brief introduction on the general structure, the properties and the functioning of a neural network, as well as the architecture, are recalled. They will be exploited for implementing the neural network fault estimators.

In this work, a set of neural network estimators is designed and trained in order to reproduce the behaviour of the systems under investigation, thus accomplishing the modelling and identification task. The structure of the i-th single neuron (Haykin, 2001) is also called *perceptron*. It features a MISO system where the output y_i is computed as a function f of the weighted sum v_i of all n_i neuron inputs $u_{i,1}, \ldots, u_{i,n_i}$ with the associated weights $w_{i,1}, \ldots, w_{i,n_i}$. The function f, called the *activation function*, represents the engine of the neuron, as shown in Fig. 6.

A structural categorisation of neural networks concerns the way in which their elements are connected with each other (Xu *et al*., 1994). In a *feed-forward network*, also called the *multilayer perceptron*, neurons are

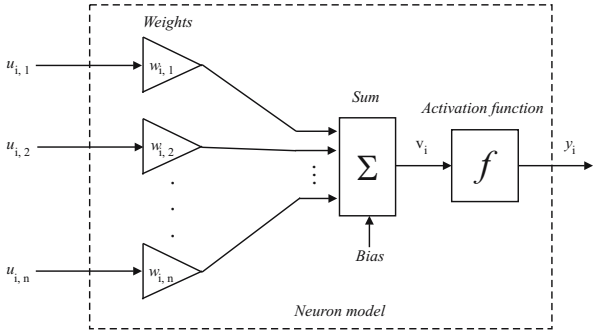

Fig. 6. i-th neuron model.

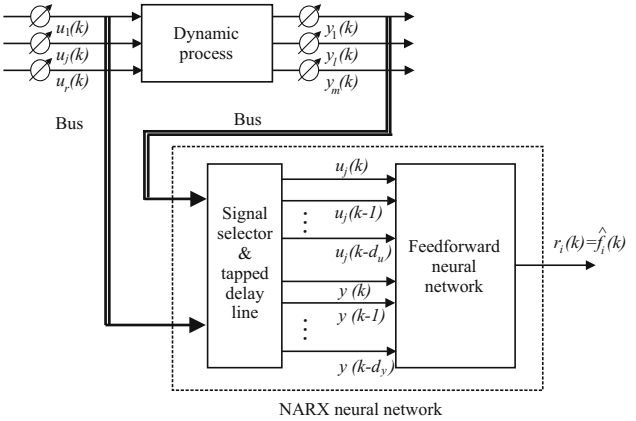

Fig. 7. NARX network used as a residual generator, i.e., the fault estimation.

grouped into unidirectional layers. The first of them, namely the *input* layer, is fed directly by the network inputs, then each successive *hidden* layer takes the inputs from the neurons of the previous layer and transmits the output to the neurons of the next layer, up to the last *output* layer, in which the final network outputs are produced. Therefore, 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, which has to be equal to the number of actual network outputs. On the other hand, *recurrent networks* (Hunt *et al*., 1992) are multilayer networks in which the output of some neurons is fed back to neurons belonging to previous layers. Thus the information flows in forward as well as in backward directions allowing a dynamic memory inside the network.

A noteworthy intermediate solution is provided by the multilayer perceptron with a tapped delay line, which is a feed-forward network whose inputs come from a delay line. This kind of network represents a suitable tool to predict the dynamic relationship between the input-output measurements and the considered fault functions. In particular, the proposed NARX network is fed by the delayed samples of the system inputs and outputs selected by the FMEA tool. Indeed, if properly trained, the NARX network can estimate the current (and the next) fault samples $f_j(k)$ on the basis of the acquired past measurements of system inputs and outputs $\mathbf{u}(k)$ and $\mathbf{y}(k)$, respectively, in the same way as in fuzzy systems.

Generally speaking, considering a MIMO system, the elaborations of the open-loop NARX network follow the law

$$\hat{f}_i(k) = f_{\text{net}}\big(\dots, u_j(k), \dots, u_j(k-d_u), \dots \\ \dots, y_l(k-1), \dots, y_l(k-d_y), \dots\big) \tag{28}$$

where $\hat{f}_i(k)$ is the estimate of the generic i-th fault, while $u_j(\cdot)$ and $y_l(\cdot)$ are the generic j-th and l-th components of the measured inputs and outputs $\mathbf{u}$ and $\mathbf{y}$, respectively, that are selected via the FMEA tool. Here k is the time step, d_u and d_y are the numbers of delay of inputs and outputs, respectively, that have to be properly estimated. Moreover, f_{net} is the function realised by the network, which depends on the layer architecture, the number of neurons, their weights and their activation functions. The behaviour of the NARX network used as the estimator of the generic fault $f_i(k)$ is depicted in Fig. 7.

The parameters on which the designer can act concerns the overall architecture (the number of neurons, connections between layers), while the values of the weights inside each neuron are derived from the network training.

3.5. Neural network training. A neural network is a learning system requiring an initial training procedure that adjusts the weights to improve the network performance. When the network task is the estimation of a nonlinear function, the training is performed by presenting to the network a set of examples of proper behaviour, consisting in the selected system inputs and outputs, $\mathbf{u}(k)$ and $\mathbf{y}(k)$, as well as the desired targets, i.e., the fault functions $\mathbf{f}(k)$. The training can be implemented in two different ways:

- *incremental mode*: each input-target pair generates an update of the network weights;
- *batch mode*: all inputs and targets are applied to the network before the weights are updated.

Although the latter kind of training requires more storage, it is characterised by faster convergence and produces smaller errors. That is why it will be considered in the following.

The training objective is the minimisation of a performance function E, which depends on the weight vector $\mathbf{w}$.

Generally speaking, for P available example patterns consisting in the input-target pairs ($\mathbf{u}_p$, $\mathbf{t}_p$), defining the output $\hat{\mathbf{y}}_p$ generated by the network fed by $\mathbf{u}_p$, the p-th

error vector can be expressed as

$$\mathbf{e}_p = [\mathbf{t}_p - \hat{\mathbf{y}}_p] = [e_{p,1}, \ldots, e_{p,M}]^T \tag{29}$$

with $p = 1, \ldots, P$ and M the number of outputs. Furthermore, the global error vector $\bar{\mathbf{e}}$ collects each $\mathbf{e}_p$:

$$\bar{\mathbf{e}} = [e_{1,1}, \ldots, e_{1,M}, \ldots, e_{P,1}, \ldots, e_{P,M}]^T. \tag{30}$$

Consequently, the performance function becomes

$$E(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P}(\mathbf{t}_i - \hat{\mathbf{y}}_i)^2 = \frac{1}{P}\sum_{p=1}^{P}\sum_{m=1}^{M} e_{p,m}^2, \tag{31}$$

where the dependence of E on the N parameters grouped in the vector $\mathbf{w} = [w_1, \ldots, w_N]^T$ is implicit in the generated output $\hat{\mathbf{y}} = \hat{\mathbf{y}}(\mathbf{w})$.

Any standard numerical optimisation algorithm can be used to update the parameters in order to minimise E. Among these, the most commons are iterative, and make use of characteristic matrices, such as the gradient $\mathbf{g}$ (or the Hessian $\mathbf{H}$) of the performance function, or the Jacobian $\mathbf{J}$ of the estimation error, defined as

$$\mathbf{g} = \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \left[\frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_N}\right]^T, \tag{32}$$

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 E}{\partial w_1^2} & \cdots & \dfrac{\partial^2 E}{\partial w_1 \partial w_N} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 E}{\partial w_N \partial w_1} & \cdots & \dfrac{\partial^2 E}{\partial w_N^2} \end{bmatrix}, \tag{33}$$

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial e_{1,1}}{\partial w_1} & \cdots & \dfrac{\partial e_{1,1}}{\partial w_N} \\ \dfrac{\partial e_{1,2}}{\partial w_1} & \cdots & \dfrac{\partial e_{1,2}}{\partial w_N} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial e_{P,M}}{\partial w_1} & \cdots & \dfrac{\partial e_{P,M}}{\partial w_N} \end{bmatrix}. \tag{34}$$

The successive iterations of these algorithms consist in updating the parameters and the calculation of the new value of the performance function, until a stop criterion is met. The updating rules of the most common optimisation algorithms (i.e., the gradient descent, Newton, Gauss–Newton and Levenberg–Marquardt algorithms) are reported in Table 3.

Table 3 summarises the parameters of the updating rules of the most common optimisation algorithms, aimed at the minimisation of the performance function E. Here k is the iteration index, α is the learning rate and μ the combination coefficient.

It can be demonstrated that the gradient descent algorithm, for a sufficiently small learning rate α value, is asymptotically convergent: around the solution $\mathbf{g}$, the gradient is close to zero and the weights practically do not change. Otherwise, the Newton and the Gauss–Newton algorithms provide faster convergence, but they both involve the computation of the inverse of a matrix which may not be invertible, causing instability in the procedure. Moreover, the Hessian matrix entails a burdensome computational effort, as it contains the second order derivative terms.

Table 3. Updating rule parameters.

Algorithm	Updating rule
Gradient descent	$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha \mathbf{g}_k$
Newton	$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{H}_k^{-1}\mathbf{g}_k$
Gauss–Newton	$\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathbf{J}_k^T\mathbf{J}_k\right)^{-1}\mathbf{J}_k\bar{\mathbf{e}}_k$
Levenberg–Marquardt	$\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathbf{J}_k^T\mathbf{J}_k + \mu\mathbf{I}\right)^{-1}\mathbf{J}_k\bar{\mathbf{e}}_k$

The *Levenberg–Marquardt* algorithm, originally proposed by Hunt *et al*. (1992), introduces an approximation of the Hessian matrix as $\mathbf{H} \approx \mathbf{J}^T\mathbf{J} + \mu\mathbf{I}$, where the first term of the sum is the Jacobian approximation (also exploited in Gauss–Newton) and the second term, driven by the combination coefficient $\mu > 0$, ensures the invertibility of the resulting matrix. Therefore, the Levenberg–Marquardt algorithm provides both fast and stable convergence and it represents a suitable tool to train a neural network. Indeed, as shown in Section 4, the neural network fault estimator blocks have been trained exploiting this method.

The training of a neural network based on the Levenberg–Marquardt algorithm, as explained by Hunt *et al*. (1992), uses a technique called *back-propagation* training, in order to compute the Jacobian matrix for the updating rule. Its name refers to the backward processing that starts from the output layer of the network towards the first layer, after a previous forward computation of neuron outputs.

3.6. Failure mode and effect analysis. The FDI schemes adopted in the simulations reported in Section 4 were already discussed in Section 3.1. However, the so-called failure mode and effect analysis (FMEA) is exploited here, as it leads to an effective design of the fault estimators, as shown below.

Following the guidelines reported by Stamatis (2003), an FMEA strategy has been performed on the wind turbine system. The FMEA is a sensitivity analysis aimed at estimating the most sensitive measurements $u_j(k)$ and $y_l(k)$ with respect to the simulated fault conditions $f_i(k)$. In practice, the monitored fault signals have been injected into the benchmark simulator, assuming that only a single fault may occur in the considered plant. Then, the relative mean square errors (RMSEs) between the fault-free and faulty measured

signals are computed, so that, for each fault, the most sensitive signal can be selected. The results of the FMEA are shown in Table 4 for the wind turbine benchmark.

In particular, the FMEA can be conducted on the basis of a selection algorithm that is achieved by introducing the normalised sensitivity function

$$N_x = \frac{S_x}{S_x^*},$$

where

$$S_x = \frac{\|x_f(k) - x_n(k)\|_2}{\|x_n(k)\|_2}, \tag{35}$$

$$S_x^* = \max_k \frac{\|x_f(k) - x_n(k)\|_2}{\|x_n(k)\|_2}. \tag{36}$$

Its value represents the effect of the considered fault case with respect to a certain measure signal $x(k)$, with $k = 1, 2, \ldots, N$. The subscripts 'f' and 'n' indicate the faulty and fault-free cases, respectively. Therefore the measurements most affected by the considered fault imply a value of N_x equal to 1. Otherwise, a small value of N_x, i.e., close to zero, denotes a signal x not affected by the fault. The signals characterised by high values of N_x can be selected as the most sensitive measurements and they will be considered in the design of the FDI blocks. The results of the FMEA sensitivity are reported in Table 5. The selected signals for each fault included in the wind turbine benchmark are divided as inputs and outputs.

As a result, the fault diagnosis blocks that have to be designed can implement the reduced fault models instead of the overall system model (7) with a noteworthy simplification of the inner structure, thus providing a decrease in the computational effort.

Note that the fault sensitivity analysis proposed in this section does not need to check all possible combinations among the input-output measurements, since the fault sensitivity indices represented by the relations (35) and (36) are performed once on a single input or output measurement. The complexity is thus simply $\mathcal{O}(r+m)$, where r is the number of inputs and m the number of outputs. This procedure can be considered logically equivalent but less complex than a structural analysis, for example, as proposed by Blanke *et al.* (2003), which may require the definition of structural residuals and the derivation of relationships based on the physical model. In fact, the model of the wind turbine considered in this work contains a relation that is expressed in the form of two-dimensional map (or a Simulink look-up table), as described by (2). In contrast, the sensitivity analysis proposed in this work is easier and straightforward, since it requires only the simulation of the considered benchmark (already available as Simulink model) where the faults are directly injected into the dynamic process. Moreover, since the benchmark is directly available as a functional scheme, it provides also a viable and direct way for the generation of the required fault sensitivity signals required by the analysis proposed in this section. In order to highlight the features of the proposed tool, an alternative approach based on a correlation analysis is outlined in the following.

In more detail, the correlation analysis is based on the following formula:

$$R_N(\tau) = \frac{1}{N-1} \sum_{k=\tau}^{N+\tau-1} (x_f(k) - x_n(k)) \times (x_f(k+\tau) - x_n(k+\tau)) \ , \tag{37}$$

with $\tau > 1$ representing the correlation window. If the signal $x_f(k) - x_n(k)$ has correlation properties, it means that the generic measurement $x(k)$ is affected by the considered fault. In fault-free conditions, the variable $R_N(\tau)$ is modelled by a Gaussian distribution for a fixed confidence interval (Ljung, 1999).

The rationale of using this correlation approach is highlighted in the following by analysing its efficacy in the achieved results. As an example, Fig. 8 reports the results of the correlation analysis on 4 measurements acquired from the wind turbine simulator. In fact, in fault-free conditions, the residuals $x_f(k) - x_n(k)$ should be ideally uncorrelated, and independent of the other measurements $x(k)$. This situation guarantees that the considered measurement $x(k)$ is not affected by the considered fault.

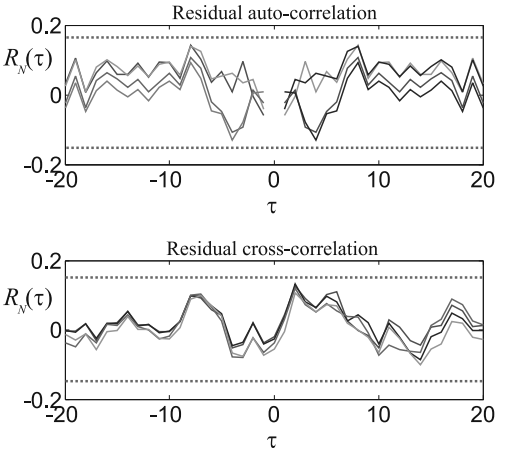

Fig. 8. Examples of auto- (top) and cross-correlation (bottom) for four measurements $x(k)$ in fault sensitivity analysis.

Figure 8 shows the example of residuals $x_f(k) - x_n(k)$ in fault-free conditions when 4 measurements are checked, thus highlighting their uncorrelation. In particular, (a) the auto-correlation function of $x_f(k) - x_n(k)$ and (b) the cross-correlation function between $x_f(k) - x_n(k)$ and $u(k)$ are displayed for 20 lags. For these variables, the 99% confidence intervals are also depicted in dotted lines, thus showing that $x_f(k) - x_n(k)$ is also independent of $x(k)$ for each of the four

Table 4. RMSE values of the most sensitive measurements $u_j(k)$ and $y_l(k)$ with respect to the faults $f_i(k)$.

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

Table 5. FMEA and the most sensitive measurements with respect to the fault cases.

Fault 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	$\beta_{1,m2}$	$\omega_{g,m1}$, $\omega_{g,m2}$

measurements. These results prove in simulation the validity of the alternative sensitivity analysis considered in this study, which leads to the same results of Table 5.

Finally, another important issue concerns the capabilities of the proposed solutions when other faults not included in the benchmarks are considered. In general, when new faults that were not previously included in the benchmark are considered, both the FMEA analysis and the residual generator estimation have to be performed again. However, this problem could be overcome by using more advanced (e.g., self-tuning) methodologies exploiting for example adaptive neural networks, (fuzzy) self-organising maps, as well as on-line estimation schemes (e.g., recursive Frisch scheme) (Beghelli *et al.*, 1990; Ioannou and Sun, 1996; Babuška, 1998; Simani and Castaldi, 2013). However, their implementations could require more complicated on-line implementations, especially when considered for real-time applications.

4. Simulation results

This section reports the simulations related to the considered benchmark, in which the proposed solutions for fault diagnosis have been implemented. Firstly, the focus is placed on the single wind turbine benchmark. Both the fuzzy and the neural network fault estimators are analysed and validated by means of an MC analysis. Then, their performances are compared with those of other fault diagnosis methods, commonly adopted in the related literature. Finally, in order to assess the proposed systems in a more realistic framework, the HIL test has been performed by means of an industrial computer interacting with on-board electronics.

4.1. Wind turbine simulations. In the following, with reference to the wind turbine benchmark model of Section 2, all the simulations are driven by the same wind mean speed sequence reported in Fig. 9. It comes from real acquisition of wind speed data. It 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. The other wind speed components are represented by uniform random variables.

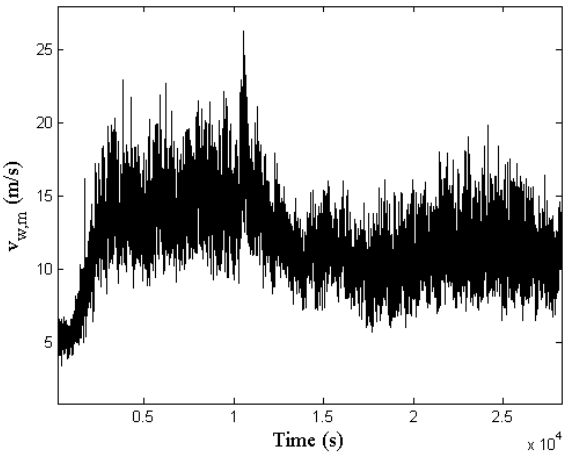

Fig. 9. Wind speed sequence driving the simulations.

The simulations last for 4400 s, during which only one fault may occur. The discrete-time benchmark model runs at a sampling frequency of 100 Hz, so that $N = 440\,000$ samples per simulation are acquired. With reference to the different scenarios described in

Section 2.4, Table 6 reports the shape and the time of the fault signals affecting the system. They are modelled as input (actuator) or output (sensor) additive faults, based on the FMEA results of Section 3.6.

Table 6. Fault characteristics.

Fault	Type	Shape	Time (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	$4100 - 4300$

In order to highlight how faults affect the system, the comparison between the faulty and the fault free signal is represented in Fig. 10 regarding the most affected signals of the FMEA test. As an example, the cases of faults 1, 2, 3, and 8 are reported here.

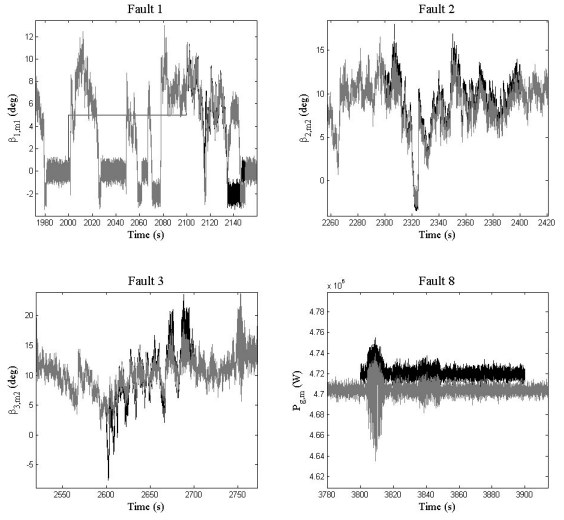

Fig. 10. Faulty signals (black line) compared with the fault-free signals (grey line).

4.2. Fault diagnosis via fuzzy identified models. The issue on the fault diagnosis of the wind turbine benchmark model via fuzzy models was proposed by Simani *et al.* (2014; 2015), but the fuzzy models were used as wind turbine output predictors, rather than fault estimators, as investigated in this paper. As addressed in Section 3.2, the fuzzy c-means clustering exploits a number $n_C = 4$ of clusters and $o = 3$ delays on input and output regressors. The algorithm generates the membership function points that are fitted through Gaussian membership functions.

Afterwards, the regressands α and δ of (16) are identified for each cluster, following the procedure explained in Section 3.2. As a result, the TS models can be implemented and the nine fault estimators are built and organised into the estimator scheme of Section 3.1, in order to accomplish the fault detection, isolation and identification tasks.

The capabilities of the fuzzy TS models used are evaluated again in terms of the root mean squared error (RMSE), where the error is calculated 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 7 shows the achieved fault prediction performances of the 9 designed fault estimators of Fig. 5.

Table 7. Fault estimator performance in terms of RMSE.

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	

In the following, an example of the structure of the fuzzy model used for the reconstruction of the fault $\hat{f}_1$ is reported. As shown in Table 5, the estimation of this fault requires the proper processing of the signals $\beta_{1,m1}$, $\beta_{1,m2}$, and $\omega_{g,m2}$, as well as the identified consequents, as summarised below for each rule:

Rule 1: $\hat{f}_1(k) = 0.97\,\hat{f}_1(k-1) - 8.93 \cdot 10^{-4}\,\hat{f}_1(k-2)$
$+4.01 \cdot 10^{-2}\,\hat{f}_1(k-3) + 9.12\,\beta_{1,m1}(k-1)$
$-11.6\,\beta_{1,m1}(k-2) + 2.46\beta_{1,m1}(k-3)$
$-0.121\,\beta_{1,m2}(k-1) + 0.34\,\beta_{1,m2}(k-2)$
$-0.21\,\beta_{1,m2}(k-3) + 4.28\,\omega_{g,m2}(k-1)$
$-0.423\,\omega_{g,m2}(k-2) - 3.77\,\omega_{g,m2}(k-3),$

Rule 2: $\hat{f}_1(k) = 1.04\,\hat{f}_1(k-1) - 3.41 \cdot 10^{-2}\,\hat{f}_1(k-2)$
$-1.21 \cdot 10^{-4}\,\hat{f}_1(k-3) - 0.23\,\beta_{1,m1}(k-1)$
$+0.32\,\beta_{1,m1}(k-2) - 0.17\,\beta_{1,m1}(k-3)$
$-1.17 \cdot 10^{-2}\,\beta_{1,m2}(k-1)$
$+2.28 \cdot 10^{-3}\,\beta_{1,m2}(k-2)$
$+1.24 \cdot 10^{-2}\,\beta_{1,m2}(k-3)$
$-0.62\,\omega_{g,m2}(k-1) + 0.12\,\omega_{g,m2}(k-2)$
$+0.54\,\omega_{g,m2}(k-3),$

Rule 3: $\hat{f}_1(k) = 0.94\,\hat{f}_1(k-1) + 3.17 \cdot 10^{-2}\,\hat{f}_1(k-2)$
$+1.8 \cdot 10^{-2}\,\hat{f}_1(k-3) - 10.1\,\beta_{1,m1}(k-1)$
$+4.08\,\beta_{1,m1}(k-2) + 6\,\beta_{1,m1}(k-3)$
$+8.74 \cdot 10^{-2}\,\beta_{1,m2}(k-1) - 0.26\,\beta_{1,m2}(k-2)$
$+0.18\,\beta_{1,m2}(k-3) - 3.23\,\omega_{g,m2}(k-1)$
$+2.37\,\omega_{g,m2}(k-2) + 0.85\,\omega_{g,m2}(k-3),$

$$\begin{aligned}\text{Rule 4:}\quad \hat{f}_1(k) &= 0.97\,\hat{f}_1(k-1)+3.01\cdot 10^{-2}\,\hat{f}_1(k-2)\\ &+1.14\cdot 10^{-2}\,\hat{f}_1(k-3)+2.97\,\beta_{1,m1}(k-1)\\ &-5.10\,\beta_{1,m1}(k-2)+1.99\,\beta_{1,m1}(k-3)\\ &+1.38\cdot 10^{-2}\,\beta_{1,m2}(k-1)\\ &-1.56\cdot 10^{-2}\,\beta_{1,m2}(k-2)\\ &-2.12\cdot 10^{-3}\,\beta_{1,m2}(k-3)\\ &+1.75\,\omega_{g,m2}(k-1)+0.35\,\omega_{g,m2}(k-2)\\ &-2.03\,\omega_{g,m2}(k-3).\end{aligned} \tag{38}$$

These estimated signals $\hat{f}_i$ are directly exploited as residuals r_i, as described by (9), and they are compared with the thresholds of (10), optimally selected in order to achieve the optimisation of the fault diagnosis performance indices, e.g., the missed fault and the false alarm rate, defined in the following. Table 8 reports the adopted δ value for the threshold logic of each fault estimator i.

Note that, as highlighted by the example of (38), in general each of the nine fuzzy fault estimators has three inputs (see Table 5), with a number of delays $n = 3$ and $n_C = 4$ clusters. Therefore, the number of estimated parameters for each fuzzy MISO model is equal to $(3 + 1) \times n = 12$. Moreover, each fault estimator requires the identification of the fuzzy membership function $\lambda_i(\cdot)$ of Eqn. (14) with $i = 1, \ldots, n_C$.

The most meaningful simulation results, proposed in the following, consider two actuator faults f_u and two sensor faults f_y, namely faults 1,4 and faults 8, 9 of the scenarios described in Section 2.4.

These faults change the monitored input and output signal $\mathbf{u}$, $\mathbf{y}$ affecting the residual $r_1 = \hat{f}_1$, $r_4 = \hat{f}_4$ and $r_8 = \hat{f}_8$, $r_9 = \hat{f}_1$ generated by the fuzzy fault estimators. These faults $\hat{f}_i$ are displayed in Fig. 11. They clearly show the achievement of the fault detection task, as they are significantly above the threshold bounds only when the relative fault is active.

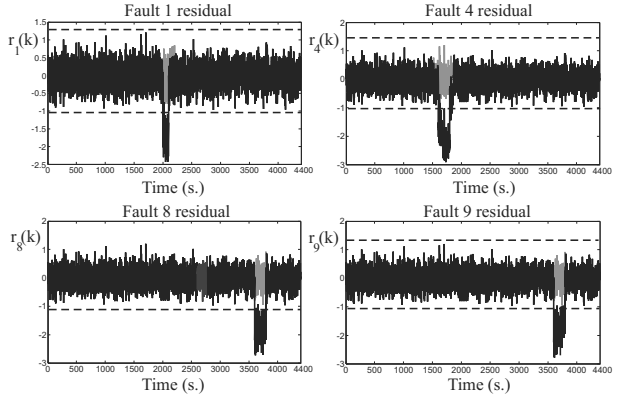

Fig. 11. Fault-free (grey line) and faulty (black continuous line) residuals for faults 1, 4, 8 and 9.

Figure 11 shows the estimated fault functions $\hat{f}_i(k)$ generated in faulty conditions by the fuzzy estimators (black continuous line) compared with the fault-free residuals (grey line). The fixed thresholds are depicted with dotted lines. The considered residuals regard 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 and the measurement errors.

Finally, as remarked at the end of Section 3.1, Fig. 11 shows an example of fault detection logic relying on time-varying thresholds, rather than the fixed ones, depicted in Fig. 11. In particular, these thresholds have been determined by means of the methodology already proposed by the same authors, (e.g., Simani and Diversi, 2003).

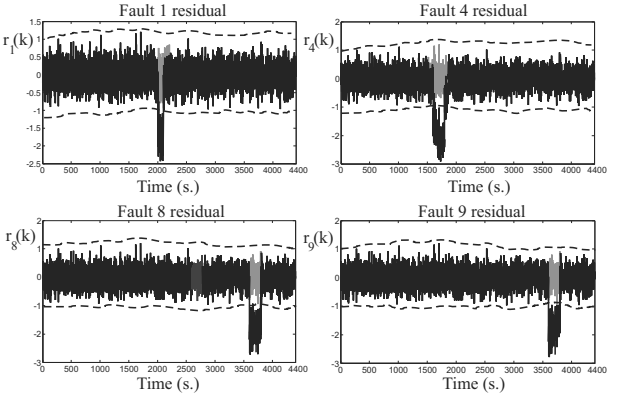

Fig. 12. Fault-free (grey line), faulty (black continuous line) residuals for faults 1, 4, 8, 9, and time-varying thresholds (dashed grey lines).

4.3. Fault diagnosis via neural networks. As for the fuzzy systems, nine NARX neural network described in Section 3.4 have been designed to estimate the nine faults affecting the acquired measurements. The selected architecture of the neural networks involves three layers, namely the input layer, the hidden layer and the output layer. The number of neurons in the input layer is 3, the number in the hidden layer has been fixed to $n_h = 16$, while the output layer has only one neuron. Finally, a number of $d_u = d_y = 4$ has been chosen for the input-output delays. Both the input and hidden layers use sigmoidal activation functions, while the output layer exploits the linear one. Each of the nine neural networks is driven by three inputs, as highlighted in Table 5.

Similarly to the fuzzy models, the neural network modelling capabilities have been tested in terms of RMSE and the results are reported in Table 9, obtained by comparing the reconstructed faults and the actual ones.

The fault detection task is achieved by comparing the residuals $r_i = \hat{f}_i(k)$ of (9) with a fixed optimised threshold, as described by (10).

Table 8. Threshold logic selection in terms of the parameter δ.

$r_i(k)$	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 9. Neural network performance in terms of RMSE.

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

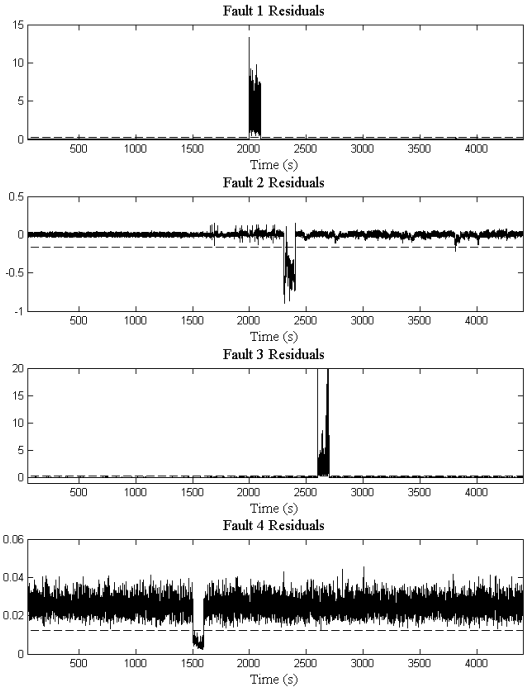

Fig. 13. Reconstructed signals (continuous line) $\hat{f}_i(k)$ and fixed thresholds (dashed line) for the actuator faults $f_1(k)$, $f_2(k)$, $f_3(k)$, $f_4(k)$.

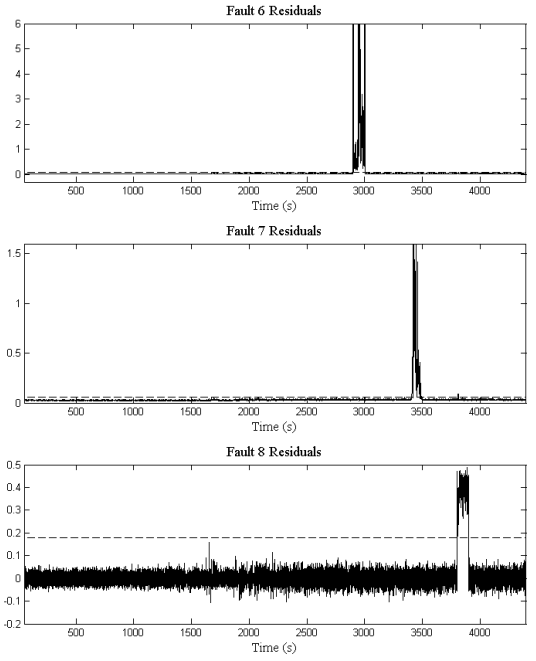

Fig. 14. Reconstructed faults $\hat{f}_i(k)$ (continuous line) and fixed thresholds (dashed line) for the sensor faults $f_6(k)$, $f_7(k)$, and $f_8(k)$.

Figure 13 shows some meaningful residual signal for actuator faults, together with the relative thresholds, while Fig. 14 shows the estimated signals regarding the sensor faults. Further details on validation and comparative results are described in the following.

In particular, Fig. 13 shows the residuals $\hat{f}_i(k)$ generated in faulty conditions by neural network estimators (continuous line) compared with the fixed thresholds (dashed line). The considered residuals concern the actuator faults $f_1(k)$, $f_2(k)$, $f_3(k)$, and $f_4(k)$.

On the other hand, Fig. 14 shows the residuals $\hat{f}_i(k)$ generated by the neural network estimators (continuous line) compared with the fixed thresholds (dashed line). The residuals considered concern the sensor fault cases $f_6(k)$, $f_7(k)$, and $f_8(k)$.

Finally, since the proposed solutions have been working quite well in the considered application, it is worth discussing advantages and disadvantages of the two approaches considered in this paper. On the one hand, the fuzzy estimation method leads to the direct estimation of the residual generator parameters (via the Frisch scheme algorithm), which is able to provide also the model orders. However, it requires the preliminary fuzzy clustering of the data used by the Frisch scheme identification approach. On the other hand, the neural network methodology does not require any data fuzzy clustering, but a trial-and-error procedure for the selection of the optimal number of neurons and

Table 10. Variations in MC parameters.

Parameter	Nominal value	Min. error	Max. error
ρ	$1.225\ \text{kg/m}^3$	$\pm 0.1\%$	$\pm 20\%$
J	$7.794 \times 10^6\ \text{kg/m}^3$	$\pm 0.1\%$	$\pm 30\%$
C_p	C_{p0}	$\pm 0.1\%$	$\pm 50\%$

delays is needed in general. In the authors' opinion, by taking into account also previous applications of the same proposed approaches (Simani, 2013), the fuzzy strategy has appeared to be more powerful and flexible general than the neural network scheme.

4.4. Validation and comparative analysis. The evaluation of the performances of the considered fault diagnosis strategies is based on the computation of the following indices:

- false alarm rate (FAR): the ratio of the number of wrongly detected faults to the number of simulated faults;
- missed fault rate (MFR): the ratio of the total number of missed faults to the number of simulated faults;
- true FDI rate (TFR): the ratio of the number of correctly detected faults to the number of simulated faults (complementary to MFR);
- mean FDI delay (MFD): the delay time of the fault occurrence to the fault detection.

A proper MC analysis has been performed in order to compute these indices and to test the robustness of the considered FDI schemes. Indeed, the MC tool is useful at these stage, as the efficacy of the diagnosis depends on both the model approximation capabilities and the measurements errors.

In particular, a set of 1000 MC runs has been executed, during which realistic wind turbine uncertainties have been considered by modelling some meaningful variables as Gaussian stochastic processes around the nominal values and with standard deviations corresponding to the realistic minimal and maximal error values of Table 10.

In addition to the proposed fuzzy and neural network fault estimators, the performance indices of other fault diagnosis schemes are analysed, as described by Odgaard and Stoustrup (2015).

The first alternative approach considered here uses a support vector machine (SVM) based on a Gaussian kernel (GKSV) developed by Laouti *et al.* (2011). The scheme defines a vector of features for each fault, which contains relevant signals obtained directly from measurements, filtered measurements or their combinations. These vectors are subsequently projected onto the kernel of the SVM, which provides suitable residuals for all of the defined faults. Data with and without faults were used for training the model for the FDI of the specific faults.

The second scheme consists in an estimation-based (EB) solution shown by Zhang *et al.* (2011). In particular, a fault detection estimator is designed to detect a fault, and an additional bank of estimators is derived to isolate them. The method was designed on the basis of a system linear model and used fixed thresholds. Each estimator for fault isolation was computed on the basis of the particular fault scenario under consideration.

The third method relying on up-down counters (UDC) was addressed by Ozdemir *et al.* (2011). These tools are commonly used in the aerospace framework, and they provide a different approach to the decision logic applied to the FDI residuals. Indeed, the decision to declare the fault occurrence involves discrete-time dynamics and is not simply a function of the residual value.

The fourth approach are combined observer and Kalman filter (COK) methods (Chen *et al.*, 2011). It relies on an observer used as a residual generator for diagnosing the faults of the drive train, in which the wind speed is considered as a disturbance. This diagnosis observer was designed to decouple the disturbance and simultaneously achieve optimal residual generation in a statistical sense. For the other two subsystems of the wind turbine, a Kalman filter-based approach was applied. The residual evaluation task used a generalised likelihood ratio test, and cumulative variance indices were applied. For fault isolation, a bank of residual generators was exploited. Sensor and system faults were thus isolated via a decision table.

Finally, the fifth method is a general fault model (GFM) scheme, which is a method of automatic design (Svard and Nyberg, 2011). The FDI strategy consists of three main steps. In the first step, a large set of potential residual generators was designed. In the second step, the most suitable residual generators to be included in the final FDI system were selected. In the third step, tests for the selected set of residual generators were performed, which were based on comparisons of the estimated probability distributions of the residuals, evaluated with fault-free and faulty data.

The comparative analysis results are reported in Table 11. In particular, different approaches to the fault

diagnosis of the wind turbine benchmark model, i.e., the fuzzy and the neural network estimators, are shown.

The results show the efficacy of the proposed FDI solutions. In details, both fuzzy and neural network estimators seem to work better than other approaches, and they have a noteworthy performance level considering the mean delay time, which is significantly lower than 10 s for all the fault cases. Also false alarm and missed fault rates are often lower than those of other approaches. Particularly neural networks features an almost null missed fault rate for all the considered faults. However, for both fuzzy and neural networks FDI design, optimisation stages are required, for example, for the selection of the optimal thresholds. Furthermore, GKSV involves delays bigger than 25 s, with false alarms and missed fault rate up to 35%. EB has comparable performance with respect to GKSV in terms of false alarm, true detection and a missed fault rates, but with a quicker detection. UDC often involves high false alarms rates, greater than 12% for all the detectable faults. COK and GFM have similar performances, with delay times higher than 10 s, false alarm and missed fault rates greater than 10%. Fault 9 concerns the drive train. This fault is difficult to detect at the wind turbine level. Therefore it is investigated also in the context of wind farm installations (Odgaard and Stoustrup, 2013). However, the fuzzy estimators can detect it, with a minimum delay but with a lower true FDI rate, with respect to the other fault cases.

4.5. Hardware-in-the-loop tests. The HIL test rig has been implemented in order to validate the proposed fault diagnosis schemes in more realistic real-time working conditions. These experimental tests aim at validating the noteworthy results obtained in simulations, considering the almost real conditions that the systems under analysis (i.e., the wind turbine in this paper) may deal with, during their working situations. This tool was originally proposed by Simani (2012) but for fault tolerant control, rather than the direct reconstruction of the fault signals aimed at fault diagnosis.

The set-up of the test-rig, represented in Fig. 15, consists of three interconnected components:

- *Simulator*: The models of the system dynamics have been implemented in LabVIEW® and consider factors such as disturbance, measurement noise and uncertainty, in addition to the system models described in Section 2. This software tool runs on an industrial CPU and allows the real-time monitoring of the simulated system parameters.

- *On board electronics*: The fault estimators have been implemented in the AWC 500 system, which features standard wind turbines specifications. This element receives the signals relative to the generated power and the generator angular rates. Then, it processes the control algorithm, including the fault diagnosis module, and produces the generator torques and pitches command signals transmitted to the simulator.

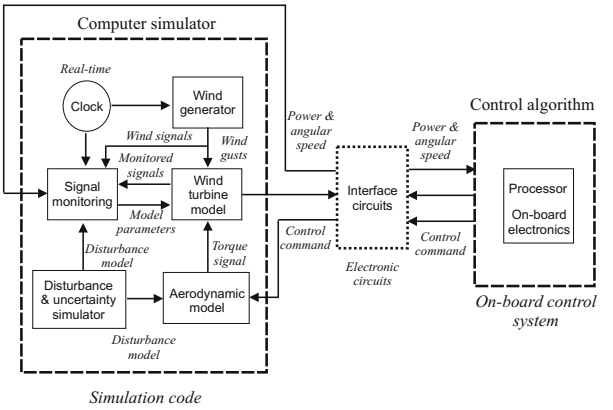

Fig. 15. Block diagram of the HIL test rig.

- *Interface circuits*: They carry out the communication between the simulator and the on-board electronics, receiving the output signals from the simulator and transmitting the signal generated by the diagnosis modules and the wind turbine controller.

Table 12 refers to the fuzzy fault diagnosis scheme and summarises the results obtained using this real-time HIL set-up for the wind turbine simulated system.

On the other hand, Table 13 refers to the neural network fault diagnosis scheme and reports the values achieved exploiting the same real-time HIL set-up used for the fuzzy fault diagnosis strategy.

It is worth observing the consistency of the near real-time test with respect to the MC analysis mentioned above in Section 4.1. Although the performances of the MC analysis seem to be better than those obtained using the HIL platform, some issues have to be taken into account. Indeed, the numerical accuracy of the on-board electronics, which involves float calculations, is more restrictive than the CPU of the simulator. Moreover, also the analog to digital and the digital to analog conversions can motivate possible deviations. Note that real situations do not require to transfer data from a computer to the on-board electronics, so that this error is not actually introduced.

However, the obtained deviations are not critical and the developed control systems can be also considered in real wind turbine applications.

5. Conclusion

The paper proposed a solution to the problem of earlier fault detection, diagnosis and isolation. The design

Table 11. Comparison of the FDI results.

Fault Case	Performance Index	GKSV	EB	UDC	COK	GFM	Fuzzy FDD	Neural FDD
1	FAR	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	MFR	0.002	0.003	0.002	0.003	0.002	0.001	0.001
	TFR	0.978	0.977	0.987	0.977	0.982	0.999	0.999
	MFD (s)	0.03	0.03	0.04	10.32	0.05	0.020	0.010
2	FAR	0.234	0.224	0.123	0.003	0.235	0.001	0.228
	MFR	0.343	0.333	0.232	0.029	0.532	0.003	0.001
	TFR	0.657	0.667	0.768	0.971	0.468	0.997	0.999
	MFD (s)	47.24	44.65	69.03	19.32	13.74	0.080	0.080
3	FAR	0.004	0.141	0.123	0.056	0.135	0.003	0.001
	MFR	0.006	0.132	0.241	0.128	0.232	0.008	0.001
	TFR	0.974	0.868	0.769	0.872	0.768	0.992	0.999
	MFD (s)	0.05	0.54	0.05	19.32	0.74	0.020	0.010
4	FAR	0.006	0.005	0.123	0.056	0.236	0.004	0.001
	MFR	0.005	0.006	0.113	0.128	0.333	0.004	0.001
	TFR	0.975	0.994	0.887	0.872	0.667	0.996	0.999
	MFD (s)	0.15	0.33	0.04	19.32	17.64	0.020	0.690
5	FAR	0.178	0.004	0.234	0.256	0.236	0.002	0.004
	MFR	0.223	0.005	0.254	0.329	0.242	0.003	0.005
	TFR	0.777	0.995	0.746	0.671	0.758	0.997	0.986
	MFD (s)	25.95	0.07	0.04	31.32	9.49	0.030	0.039
6	FAR	0.897	0.173	0.334	0.156	0.096	0.042	0.001
	MFR	0.987	0.234	0.257	0.129	0.042	0.033	0.001
	TFR	0.013	0.766	0.743	0.871	0.958	0.967	0.999
	MFD (s)	95.95	11.37	12.94	34.02	9.49	3.030	0.010
7	FAR	0.899	0.044	0.134	0.134	0.123	0.047	0.676
	MFR	0.899	0.035	0.121	0.101	0.098	0.023	0.001
	TFR	0.101	0.965	0.879	0.899	0.902	0.977	0.999
	MFD (s)	99.95	26.17	13.93	35.01	29.79	5.070	6.870
8	FAR	0.004	0.045	0.144	0.109	0.099	0.003	0.466
	MFR	0.007	0.011	0.101	0.032	0.124	0.002	0.001
	TFR	0.993	0.989	0.899	0.968	0.876	0.998	0.999
	MFD (s)	0.07	0.08	0.09	0.06	8.94	0.050	0.200
9	FAR	–	–	–	–	–	0.134	0.101
	MFR	–	–	–	–	–	0.165	0.123
	TFR	–	–	–	–	–	0.835	0.802
	MFD (s)	–	–	–	–	–	0.301	0.379

Table 12. FDI performance indices for the wind turbine HIL test with the fuzzy fault estimators.

Estimated fault $\hat{f}_i(k)$	FAR	MFR	TFR	MFD
1	0.005	0.005	0.995	0.077
2	0.004	0.004	0.996	0.490
3	0.004	0.004	0.996	0.080
4	0.005	0.005	0.995	0.070
5	0.003	0.004	0.997	0.060
6	0.004	0.005	0.996	0.760
7	0.005	0.004	0.995	0.640
8	0.005	0.004	0.995	0.060
9	0.004	0.005	0.996	0.180

Table 13. FDI performance indices for the wind turbine HIL test with the neural network fault estimators.

Estimated fault $\hat{f}_i(k)$	FAR	MFR	TFR	MFD
1	0.007	0.006	0.899	0.014
2	0.234	0.005	0.867	0.516
3	0.004	0.004	0.914	0.080
4	0.005	0.005	0.922	0.070
5	0.006	0.007	0.905	0.097
6	0.005	0.006	0.989	0.871
7	0.701	0.007	0.981	6.987
8	0.498	0.008	0.987	0.289
9	0.197	0.176	0.798	0.399

of the fault indicator, in this work represented by the direct estimate of the fault itself, involved two data-driven approaches, as they represented an effective tool for coping with a poor analytical knowledge of the system dynamics, together with noise and disturbances. In particular, the proposed data-driven solutions were based on fuzzy systems and neural networks used to describe the strongly nonlinear relationships between the input-output measurement and the considered faults. The chosen architectures belong to the nonlinear autoregressive-with-exogenous-input system topology, as it can represent a dynamic evolution of the system along time.

The developed fault diagnosis schemes were tested by means of a high-fidelity benchmark model that simulated the normal and the faulty behaviour of a wind turbine system. The achieved performances were compared with those of other control strategies, coming from the related literature. Finally, Monte-Carlo analysis and an hardware-in-the-loop test-rig served to analyse the robustness and the reliability of the proposed solutions against realistic and typical parameter uncertainties and disturbances. Further works will address the analysis of the performance of the developed fault diagnosis strategies when used for active fault tolerant control applications.

References

Babuška, R. (1998). *Fuzzy Modeling for Control*, Kluwer Academic Publishers, Boston, MA.

Beghelli, S., Guidorzi, R.P. and Soverini, U. (1990). The Frisch scheme in dynamic system identification, *Automatica* **26**(1): 171–176.

Bezdek, J.C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA.

Blanke, M., Kinnaert, M., Lunze, J. and Staroswiecki, M. Schröder, J. (2003). *Diagnosis and Fault-Tolerant Control*, 1st Edn., Springer, Berlin.

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

Castaldi, P., Mimmo, N. and Simani, S. (2017). Avionic air data sensors fault detection and isolation by means of singular perturbation and geometric approach, *Sensors* **17**(10): 1–19, DOI: 10.3390/s17102202.

Chen, J. and Patton, R.J. (1999). *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publishers, Boston, MA.

Chen, W., Ding, S.X., Sari, A.H.A., Naik, A., Khan, A.Q. and S., Y. (2011). Observer-based FDI schemes for wind turbine benchmark, *Proceedings of the 18th IFAC World Congress 2011, Milan, Italy*, Vol. 18, pp. 7073–7078, DOI: 10.3182/20110828-6-IT-1002.03469.

Dolan, D.S.L. and Lehn, P.W. (2006). Simulation model of wind turbine 3p torque oscillations due to wind shear and tower shadow, *IEEE Transactions on Energy Conversion* **21**(3): 2050–2057, DOI: 10.1109/TEC.2006.874211.

Fantuzzi, C. and Rovatti, R. (1996). On the approximation capabilities of the homogeneous Takagi–Sugeno model, *Proceedings of the 5th IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA*, pp. 1067–1072.

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

Gong, X. and Qiao, W. (2013). Bearing fault diagnosis for direct-drive wind turbines via current-demodulated signals, *IEEE Transactions on Industrial Electronics* **60**(8): 3419–3428, DOI: 10.1109/TIE.2013.2238871.

Graaff, A.J. and Engelbrecht, A.P. (2012). Clustering data in stationary environments with a local network neighbourhood artificial immune system, *International Journal of Machine Learning and Cybernetics* **3**(1): 1–26, DOI: 10.1007/s13042–011–0041–0.

Hassanabadi, A.H., Shafiee, M. and Puig, V. (2016). Robust fault detection of singular LPV systems with multiple time-varying delays, *International Journal of Applied Mathematics and Computer Science* **26**(1): 45–61, DOI: 10.1515/amcs-2016-0004.

Haykin, S. (2001). *Kalman Filtering and Neural Networks*, Wiley-Interscience, New York, NY.

Hunt, K., Sbarbaro, D., Zbikowki, R. and Gawthrop, P. (1992). Neural networks for control system: A survey, *IEEE Transactions on Neural Networks* **28**(6): 1083–1112.

Ioannou, P. and Sun, J. (1996). *Robust Adaptive Control*, Prentice-Hall, Upper Saddle River, NJ.

Jain, A. and Dubes, R. (1988). *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ.

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

Laouti, N., Sheibat-Othman, N. and Othman, S. (2011). Support vector machines for fault detection in wind turbines, *Proceedings of the 18th IFAC World Congress 2011, Milan, Italy,* Vol. 18, pp. 7067–7072, DOI: 10.3182/20110828-6-IT-1002.02560.

Ljung, L. (1999). *System Identification: Theory for the User*, 2nd Edn., Prentice Hall, Englewood Cliffs, NJ.

Odgaaard, P.F. and Shafiei, S.E. (2015). Evaluation of wind farm controller based fault detection and isolation, *Proceedings of the IFAC SAFEPROCESS Symposium 2015, Paris, France*, Vol. 48, pp. 1084–1089, DOI: 10.1016/j.ifacol.2015.09.671.

Odgaard, P.F. (2012). FDI/FTC wind turbine benchmark modelling, *in* R.J. Patton (Ed.), *Workshop on Sustainable Control of Offshore Wind Turbines*, Vol. 1, University of Hull, Hull, pp. 1–5.

Odgaard, P.F. and Stoustrup, J. (2012). Results of a wind turbine FDI competition, *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2012, Mexico City, Mexico*, Vol. 8, pp. 102–107, DOI: 10.3182/20120829-3-MX-2028.00015.

Odgaard, P.F. and Stoustrup, J. (2013). Fault tolerant wind farm control—a benchmark model, *Proceedings of the IEEE Multiconference on Systems and Control, MSC2013, Hyderabad, India*, pp. 1–6.

Odgaard, P.F. and Stoustrup, J. (2015). A benchmark evaluation of fault tolerant wind turbine control concepts, *IEEE Transactions on Control Systems Technology* **23**(3): 1221–1228.

Odgaard, P.F., Stoustrup, J. and Kinnaert, M. (2013). Fault-tolerant control of wind turbines: A benchmark model, *IEEE Transactions on Control Systems Technology* **21**(4): 1168–1182, DOI: 10.1109/TCST.2013.2259235.

Ozdemir, A.A., Seiler, P. and Balas, G.J. (2011). Wind turbine fault detection using counter-based residual threshold-ing, *Proceedings of the 18th IFAC World Congress 2011, Milan, Italy*, Vol. 18, pp. 8289–8294, DOI: 10.3182/20110828-6-IT-1002.01758.

Parker, M.A., Chong, H.N. and Ran, L. (2011). Fault-tolerant control for a modular generator-converter scheme for direct-drive wind turbines, *IEEE Transactions on Industrial Electronics* **58**(1): 305–315.

Patton, R.J., Uppal, F.J., Simani, S. and Polle, B. (2008). Reliable fault diagnosis scheme for a spacecraft attitude control system, *Journal of Risk and Reliability* **222**(2): 139–152, DOI: 10.1243/1748006XJRR98.

Patton, R.J., Uppal, F.J., Simani, S. and Polle, B. (2010). Robust FDI applied to thruster faults of a satellite system, *Control Engineering Practice* **18**(9): 1093–1109, DOI: 10.1016/j.conengprac.2009.04.011.

Rovatti, R. (1996). Takagi–Sugeno models as approximators in Sobolev norms: The SISO case, *5th IEEE International Conference on Fuzzy Systems, New Orleans, LO, USA*, Vol. 2, pp. 1060–1066.

Rovatti, R., Fantuzzi, C. and Simani, S. (2000). High-speed DSP-based implementation of piecewise-affine and piecewise-quadratic fuzzy systems, *Signal Processing Journal* **80**(6): 951–963, DOI: 10.1016/S0165-1684(00)00013-X.

Simani, S. (2012). Application of a data-driven fuzzy control design to a wind turbine benchmark model, *Advances in Fuzzy Systems* **2012** : 1–12, DOI: 10.1155/2012/504368.

Simani, S. (2013). Residual generator fuzzy identification for automotive diesel engine fault diagnosis, *International Journal of Applied Mathematics and Computer Science* **23**(2): 419–438, DOI: 10.2478/amcs-2013-0032.

Simani, S. and Castaldi, P. (2013). Data-driven and adaptive control applications to a wind turbine benchmark model, *Control Engineering Practice* **21**(12): 1678–1693, DOI: dx.doi.org/10.1016/j.conengprac.2013.08.009.

Simani, S. and Castaldi, P. (2014). Active actuator fault tolerant control of a wind turbine benchmark model, *International Journal of Robust and Nonlinear Control* **24**(8–9): 1283–1303, DOI: 10.1002/rnc.2993.

Simani, S. and Castaldi, P. (2018). Robust control examples applied to a wind turbine simulated model, *Applied Sciences* **8**(1): 1–28, DOI: 10.3390/app8010029.

Simani, S. and Diversi, R. (2003). Residual generation and identification for dynamic processes, *5th Symposium on Fault Detection Supervision and Safety for Technical Processes, SAFEPROCESS 2003, Washington, DC, USA*, Vol. 1, pp. 375–380.

Simani, S., Farsoni, S. and Castaldi, P. (2014). Fault diagnosis of a wind turbine benchmark via identified fuzzy models, *IEEE Transactions on Industrial Electronics* **62**(6): 3775–3782, DOI: 10.1109/TIE.2014.2364548.

Simani, S., Farsoni, S. and Castaldi, P. (2015). Wind turbine simulator fault diagnosis via fuzzy modelling and identification techniques, *Sustainable Energy, Grids and Networks* **1**(1): 45–52, DOI: 10.1016/j.segan.2014.12.001.

Simani, S. and Turhan, C. (2017). Adaptive signal processing strategy for a wind farm system fault accommodation, *Proceedings of the Intelligent Systems Conference, IntelliSys 2017, London, UK*, pp. 1–8.

Stamatis, D.H. (2003). *Failure Mode and Effect Analysis: FMEA from Theory to Execution*, 2nd Edn., ASQ Quality Press, Milwaukee, WI.

Svard, C. and Nyberg., M. (2011). Automated design of an FDI system for the wind turbine benchmark, *Proceedings of the 18th IFAC World Congress 2011, Milan, Italy*, Vol. 18, pp. 8307–8315, DOI: 10.3182/20110828-6-IT-1002.00618.

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

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

Xu, J.-X., Liu, C. and Hang, C. (1994). Combined adaptive and fuzzy control using multiple models, *3rd IEEE International Conference on Fuzzy Systems, Orlando, FL, USA*, pp. 22–29.

Zhang, X., Zhang, Q., Zhao, S., Ferrari, R. M.G., Polycarpou, M.M. and Parisini, T. (2011). Fault detection and isolation of the wind turbine benchmark: An estimation-based approach, *Proceedings of the 18th IFAC World Congress 2011, Milan, Italy*, Vol. 18, pp. 8295–8300, DOI: 10.3182/20110828-6-IT-1002.02808.

Silvio Simani was born in Ferrara in 1971. He received the *Laurea* degree *(cum laude)* in electrical engineering in 1996 from the Department of Engineering at the University of Ferrara (Italy), and in 2000 his PhD in information science (automatic control) at the University of Ferrara and Modena (Italy). Since 2006 he has been a senior member of IEEE, and since 2000 a member of the SAFEPROCESS Technical Committee. Since 2002 he has been an assistant professor at the Department of Engineering, University of Ferrara. His research interests include fault diagnosis, fault tolerant control, and system identification. He is an author of about 250 refereed journal and conference papers, as well as 4 books on these topics.

Saverio Farsoni was born in Mirandola (MO, Italy) in 1987. In 2012 he graduated (*cum laude*) in informatics and automation engineering from the University of Ferrara with an MSc thesis on simulations in bio-medical environments. In 2016 he was awarded a PhD in engineering science (automatic control) at the University of Ferrara (Italy). Since 2016 he has been a research fellow under the supervision of Dr. Simani. He works on control systems, fuzzy logic, modelling and identification problems. In particular, his research interests lie in fault diagnosis and fault tolerant control for eolic plants.

Paolo Castaldi was born in Bologna, Italy. He received the *Laurea* degree (*cum laude*) in electronic engineering in 1990 and the PhD degree in systems engineering in 1994, both from the Universities of Bologna, Padova and Firenze (Italy). Since 1995 he has been an assistant professor at the Department of Electronics, Computer Science and Systems of the University of Bologna. Since 2009 he has been a member of the IFAC Technical Committee on Aerospace. His research interests include fault diagnosis and fault tolerant control, adaptive filtering, system identification, and their applications to aerospace and wind turbine. He is an author of about 100 refereed journal and conference papers, as well as one book on these topics. He is an associate editor of *Control Engineering Practice*.

Received: 15 March 2017
Revised: 28 September 2017
Re-revised: 12 January 2018
Accepted: 13 January 2018