Article

Intelligent Fault Diagnosis Techniques Applied to an Offshore Wind Turbine System**

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

[1]Dipartimento di Ingegneria, Università degli Studi di Ferrara. Via Saragat 1E, Ferrara (FE) 44122, Italy. silvio.simani@unife.it

[2]Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi" – DEI, Alma Mater Studiorum Università di Bologna. Viale Risorgimento 2, 40136, Bologna (BO), Italy. paolo.castaldi@unibo.it

* Corresponding author. Correspondence: silvio.simani@unife.it; Tel.: +39-0532-97-4844

**Submitted for the Special Issue "Offshore Wind Energy". Guest Editor Dr. Mohsen N. Soltani (Aalborg University). Academic Editor: xxx

Version December 24, 2018 submitted to Entropy; Typeset by LaTeX using class file mdpi.cls

Abstract: The fault diagnosis of wind turbine systems represent a challenging issue, especially for offshore installations, thus justifying the research topics developed in this work. Therefore, this paper addresses the problem of the fault diagnosis of wind turbines, and it present viable solutions of fault detection and isolation techniques. The design of the so–called fault indicator consists of its estimate, which involves data–driven methods, as they result effective tools for managing partial analytical knowledge of the system dynamics, together with noise and disturbance effects. In particular, the suggested data–driven strategies exploit fuzzy systems and neural networks that are employed to determine nonlinear links between measurements and faults. The selected architectures are based on nonlinear autoregressive with exogenous input prototypes, as they approximate the dynamic evolution of the system along time. The designed fault diagnosis schemes are verified via a high–fidelity simulator, which describes the normal and the faulty behaviour of an offshore wind turbine plant. Finally, by taking into account the presence of uncertainty and disturbance implemented in the wind turbine simulator, the robustness and the reliability features of the proposed methods are also assessed. This aspect is fundamental when the proposed fault diagnosis methods have to be applied to offshore installations.

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

1. Introduction

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

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

These issues have motivated the development of fault diagnosis techniques that can be coupled with the fault tolerant controllers (the so–called 'sustainable' systems). On the other hand, many turbine manufacturers adopt conservative approaches against faults, which lead to the shutdown of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2018 doi:10.20944/preprints201812.0265.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

the plant in order to wait for O&M service. Hence, effective tools for coping with faults have to be investigated, in order to improve wind turbine features, particularly during faulty situations. This will lead to prevent critical failures that may affect other wind turbine components, thus avoiding unplanned replacement of functional parts, as well as the decrease of O&M costs, with the increase of the energy production. Moreover, the development of digital control systems, big data tools, and artificial intelligence strategies enhance the development of new real–time condition monitoring, diagnosis and fault tolerant control strategies for industrial processes, which can be available only on–demand.

In recent years, many works have been proposed on the topics of fault diagnosis of wind turbines, as shown very recently in [2,3]. Some of them are focused on the diagnosis of particular faults, *e.g.* those affecting the drive–train system at a wind turbine level. Sometimes these faults are better managed when the wind turbine system is considered in comparison to other parts of the whole plant [4]. Moreover, fault tolerant control of wind turbines has been investigated *e.g.* in [5] and international cooperations on these problems were also proposed [6].

Fault diagnosis oriented to the sustainability feature when applied to safety–critical systems such as wind turbines has been proven to be a challenging issue [7,8], thus motivating the research topics addressed in this paper.

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

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

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

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

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

2. Offshore Wind Turbine Simulator

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

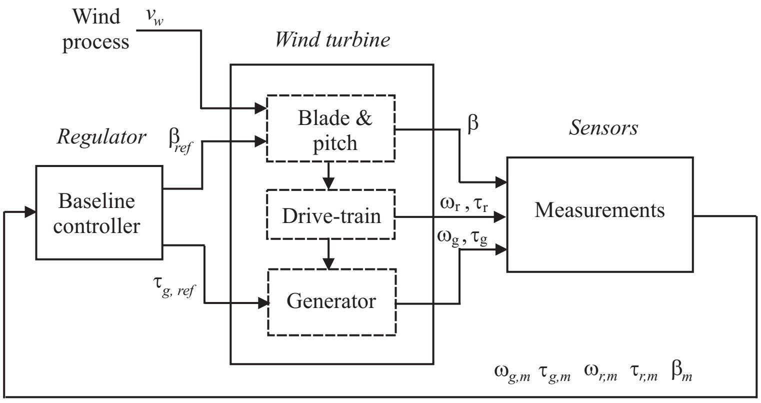

Figure 1. Scheme of the offshore wind turbine simulator.

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

2.1. Wind Turbine Model

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

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

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

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

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

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

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2018 doi:10.20944/preprints201812.0265.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

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

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

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

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

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

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

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

2.2. Simulated Fault Scenario

The wind turbine simulator includes the generation of three different typical fault cases, *i.e.* sensor, actuator and system faults [4,12].

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

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

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

These 9 fault cases are summarised in Table 1, which highlights also which measured signals are affected by them, as shown in Figure 1.

Table 1. Fault scenario of the wind turbine simulator.

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2018 doi:10.20944/preprints201812.0265.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

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

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

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

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

On the other hand, the input vector:

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

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

3. Intelligent Fault Diagnosis Techniques

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

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

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

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

Among the possible approaches exploited for residual generation, and based on the analytical redundancy principle, this work proposes to exploit fuzzy system and neural network structures, which provide an on–line estimation $\hat{\mathbf{f}}(k)$ of the fault signals $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$. Hence, as shown in Fig. 2, the so–called diagnostic residuals $\mathbf{r}(k)$ are equal to the estimated fault signals, $\hat{\mathbf{f}}(k)$, which are computed by the general fault estimator, as highlighted by Eq. (8):

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

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

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

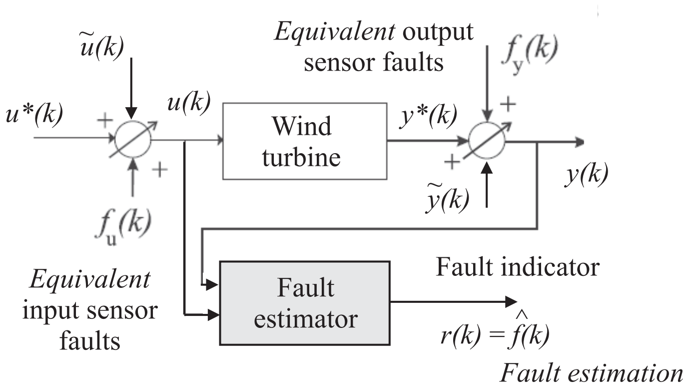

Figure 2. The residual generation scheme for fault diagnosis based on fault estimators.

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

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

with $r_i(k)$ representing the i–th component of the vector $\mathbf{r}(k)$. If it is considered as a random variable, its mean $\bar{r}_i$ and variance $\sigma^2_{r_i}$ values can be estimated in fault–free condition, after the acquisition of N samples, according to Eqs. (10):

$$\begin{cases} \bar{r}_i & = & \frac{1}{N}\sum_{k=1}^{N} r_i(k) \\ \sigma^2_{r_i} & = & \frac{1}{N}\sum_{k=1}^{N} \left(r_i(k) - \bar{r}_i\right)^2 \end{cases} \tag{10}$$

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

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

According to the scheme depicted in Fig. 3, in order to uniquely isolate one of the input or output faults, under the assumption that multiple faults cannot occur, a bank of Multi–Input Single–Output (MISO) fault estimators is design. In general, the number of this estimators is equal to the number of faults that have to be diagnosed, *i.e.* which coincides to the number of input and output measurements, $r + m$. Therefore, the i–th estimator providing the reconstruction of the fault $\hat{f}(k) = r_i(k)$ is driven by the components of the input and output signals $\mathbf{u}(k)$ and $\mathbf{y}(k)$. These components are selected in order to be sensitive to the specific fault $f_i(t)$. In fact, the design of these fault estimators is enhanced by the fault sensitivity analysis described in Section 3.3. For each case, the fault modes and their resulting effects on the rest of the system are analysed, and in particular the most sensitive input $u_j(k)$ and output $y_l(k)$ measurements to that specific fault situation are selected. In this way, by means of the fuzzy system and neural network tools, it will be possible to derive the dynamic relationships between the input–output measurements, $u_j(k)$ and $y_l(k)$, and the faults $f_i(t)$, as highlighted by Figure 3.

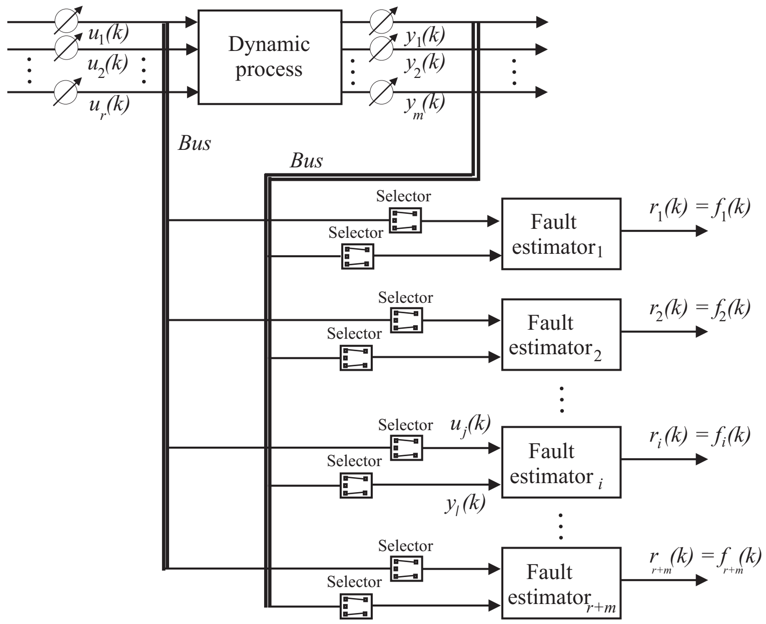

Figure 3. The estimator scheme for the reconstruction of the equivalent input or output fault, $f_i(t)$.

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

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

3.1. Fuzzy Modelling and Identification

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

According to this approach, the approximation of nonlinear Multi–Input Single–Output (MISO) systems can be achieved by the Takagi–Sugeno (TS) fuzzy reasoning, as described in [9]. The TS modelling approach proposed here, as addressed in [14], describes the consequents as deterministic functions $g_i(\cdot)$ of the inputs, while the antecedents remain fuzzy propositions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

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

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

3.2. Neural Network Modelling and Training

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

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

A categorisation of these learning structures concerns the way in which their neurons are connected each others [19]. This work proposes to use feed–forward network, also called multilayer perceptron, where the neurons are grouped into unidirectional layers. The first of them, the input layer, is directly fed by the network inputs; then, an hidden layer takes the inputs from the neurons of the input layer and transmits them the output to the neurons of the third layer, the output layer, which produces the final network outputs. According to this structure, neurons are connected from one layer to the next, but not within the same layer. The only constraint is the number of neurons in the output layer, that has to be equal to the number of actual network outputs. On the other hand, recurrent networks are multilayer networks, in which the output of some neurons is fed back to neurons belonging to previous layers, thus the information flow in forward as well as in backward directions, allowing a dynamic memory inside the network [20].

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 study proposes to use this solution, defined as quasi–static neural network, as it represents a suitable tool to predict dynamic relationships between the input–output measurements and the considered fault functions. In this way, another NARX description is obtained, since the nonlinear (static) network is fed by the delayed samples of the system inputs and outputs selected by the fault sensitivity analysis tool described in Section 3.3. Indeed, if properly trained, the NARX network can estimate the current (and the next) fault samples $f_j(k)$ on the basis of the selected past measurements of system inputs and outputs $u_l(k)$ and $y_j(k)$, respectively, in the same way of the fuzzy systems.

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

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2018 doi:10.20944/preprints201812.0265.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

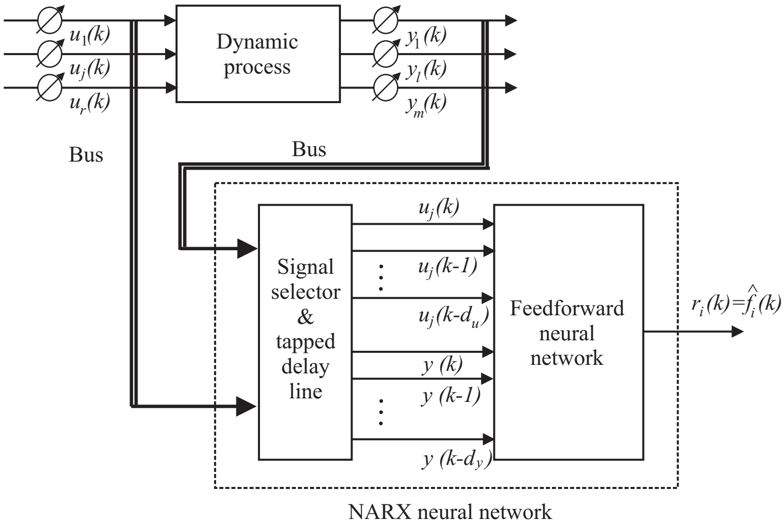

Figure 4. NARX network used as i–th residual generator, *i.e.* the fault estimator $\hat{f}(k) = r_i(k)$.

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

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

3.3. Fault Sensitivity Analysis

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

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

Table 2. The most sensitive measurement $u_j(k)$, $y_l(k)$ and the RMSE values with respect to the faults $f_i(k)$.

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2018 doi:10.20944/preprints201812.0265.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

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

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

with:

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

and:

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

The value of N_x indicates the effect of the considered fault case with respect to the general measured signal $x(k)$, with $k = 1, 2, \ldots, N$. The subscripts 'f' and 'n' indicate the faulty and the fault–free case, respectively. Therefore, the measurements that are most affected by the considered fault lead to a value of N_x equal to 1. Otherwise, a smaller value of N_x, *i.e.* close to zero, represents a signal $x(k)$ not affected by the fault. Those signals characterised by high value of N_x are thus selected as the most sensitive measurements, and they will be considered in the design of the fault diagnosis modules of the bank sketched in Figure 3.

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

Table 3. The most sensitive measurements with respect to the considered fault scenario.

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

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

4. Results and Discussion

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

With reference to the wind turbine benchmark of Section 2, all simulations are driven by the same wind mean speed sequence. It was acquired from a real measurement of wind speed, which represents a good coverage of typical operating conditions, as it ranges from 5 to 20 m/s, with a few spikes at 25 m/s [12]. The simulations last for 4400 s, with single fault occurrences. The discrete–time simulator runs at a sampling frequency of 100 Hz, so that $N = 440000$ samples are acquired during each simulation. With reference to the different fault cases reported in Section 2.2, Table 4 shows the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2018 doi:10.20944/preprints201812.0265.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

shape and the timing of the fault modes affecting the process. They model input (actuator) or output (sensor) additive faults, which are used for sensitivity analysis of Section 3.3.

Table 4. Fault modes of the wind turbine simulator.

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

As an example, in order to highlight the effective faults affect on the process measurements, Fig. 5 compares the results of the fault sensitivity test in terms of fault–free and faulty signals. The cases of the faults 1, 2, 3, and 8 are considered.

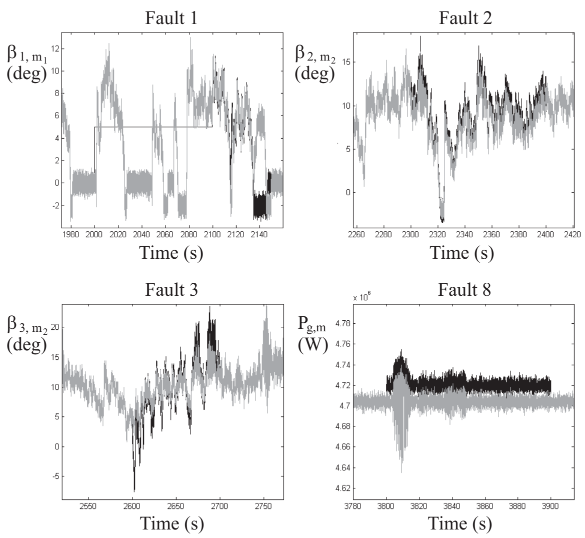

Figure 5. The fault–free (grey line) signals with respect to the faulty ones (black line).

4.1. Fault Diagnosis via Fuzzy Estimators

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

Section 3.1 suggested to exploit the fuzzy c–means clustering algorithm. When applied to the data of the wind turbine simulator, a number $n_C = 4$ of clusters and $o = 3$ delays on input and output regressors were determined. The tool also generated the membership function points, that are fitted through Gaussian membership functions. After the data clustering, the regressands $\alpha_j^{(i)}$ and $\delta_j^{(i)}$ of Eq.(15) were identified for each cluster by following the procedure of Section 3.1. The TS models of Eq. (13) were thus implemented and 9 fault estimators were designed built and organised into the estimator scheme in order to accomplish the fault diagnosis task, as sketched in Figure 3.

The effectiveness of the fuzzy TS fault estimators used were assessed in terms of Root Mean Squared Error (RMSE), which is computed as the difference between the predicted $\hat{f}_i(k)$ and the actual

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2018 doi:10.20944/preprints201812.0265.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

fault $f_i(k)$ signals for each of the fuzzy estimators, with $i = 1, \ldots, 9$. Table 5 summarises the achieved performance of the 9 fault estimators of Figure 3.

Table 5. 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 this case, these estimated signals $\hat{f}_i$ are directly exploited as diagnostic residuals r_i, as remarked by Eq. (8). They can be compared with the thresholds of Eq. (9), optimally selected in order to achieve the optimisation of the overall fault diagnosis performance indices, in terms of missed fault and the false alarm rates [22]. In particular, Table 6 summarises the values of the parameter δ of Eq. (9) for each fault estimator i.

Table 6. Threshold logic selection in terms of the parameter δ.

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

Note that in general each of the 9 fuzzy fault estimators described by the relations of Eqs. (13) and (14) has 3 inputs (see Table 3), with a number of delays $n = 3$ and $n_C = 4$ clusters. Therefore, the number of estimated parameters for each fuzzy MISO model (3 inputs and 1 output) is equal to $(3 + 1) \times n = 12$. Moreover, for each fault estimator, the estimation of the fuzzy membership functions $\lambda_i(\cdot)$ of Eq. (13) with $i = 1, \ldots, n_C$ was required.

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

According to Table 3, these faults caused the alteration of the monitored input and output signal $\mathbf{u}$, $\mathbf{y}$ affecting the residual $r_1 = \hat{f}_1$, $r_4 = \hat{f}_4$, $r_8 = \hat{f}_8$, and $r_9 = \hat{f}_9$ generated by the fuzzy fault estimators. These faults $\hat{f}_i$ depicted in Fig. 6 demonstrate the achievement of the fault diagnosis task, as they exceed the threshold levels only when the relative fault is active, as recalled in Table 4.

Figure 6 depicts the reconstructed fault functions $\hat{f}_i(k)$ generated by the fuzzy estimators in faulty conditions (black continuous line) with respect to the fault–free residuals (grey line). The fixed thresholds are depicted with dotted lines. The considered residuals refer to the fault cases 1, 4, 8 and 9. It is worth noting that in fault–free conditions the estimated fault functions $\hat{f}_i(k)$ are not zero due to both the model disturbance and the measurement errors simulated by the wind turbine benchmark This point highlights also the robustness and reliability features of the developed fault diagnosis technique relying on the proposed fuzzy tool.

4.2. Fault Diagnosis via Neural Networks

As for the fuzzy systems, 9 NARX neural network described in Section 3.2 were designed to estimate the 9 faults affecting the acquired measurements, according to the scheme of Figure 3. The neural networks selected for fault diagnosis purpose consist of 3 layers, with 3 neurons in the input layer, 16 in the hidden one, and one neuron in the output layer. A number of $d_u = d_y = 4$ delays was selected in the relation of Eq. (16). Both the input and the hidden layers used sigmoidal activation functions, whilst the output layer exploits the linear one. According to Table 3 and Figure 4, each of the 9 neural networks was driven by 3 inputs.

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

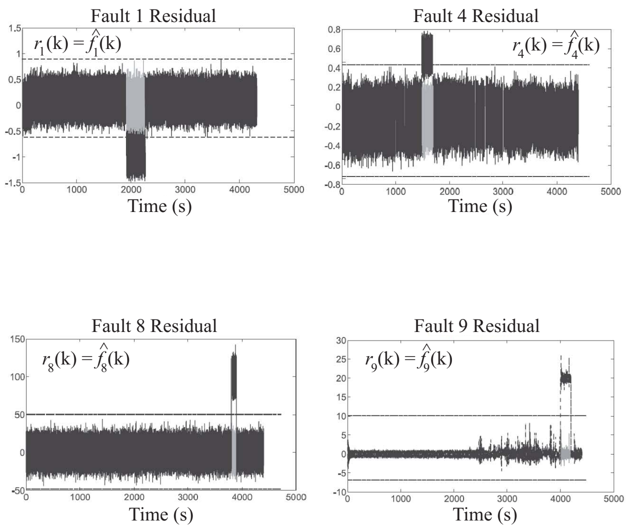

Figure 6. Fault–free (grey line) and faulty (black continuous line) residuals regarding the fault cases 1, 4, 8, and 9.

Table 7. 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	

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

Table 8. δ values for the threshold logic.

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

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

In particular, Figure 7 depicts the residuals $\hat{f}_i(k)$ generated in faulty conditions by the neural network estimators (continuous line) compared with the fixed thresholds (dashed line). The considered residuals refer to the faults $f_1(k)$, $f_2(k)$, $f_3(k)$, and $f_4(k)$ of Table 4.

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

4.3. Hardware–In–The–Loop Experiments

The Hardware–In–the–Loop (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 previous results achieved in simulations, considering more realistic conditions that the systems under diagnosis may deal with. This tool was originally proposed in [23] but for fault tolerant control purpose, rather than the fault diagnosis purpose.

Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

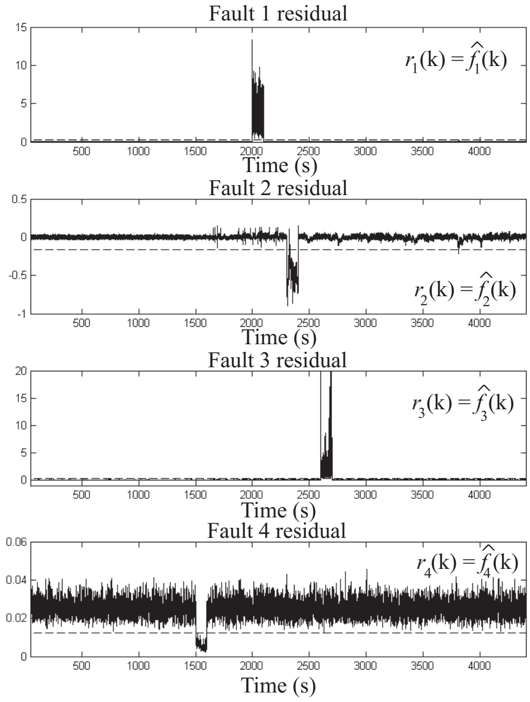

Figure 7. Estimated signals (continuous line) $\hat{f}_i(k)$ and fixed thresholds (dashed line) for the faults 1, 2, 3, and 4.

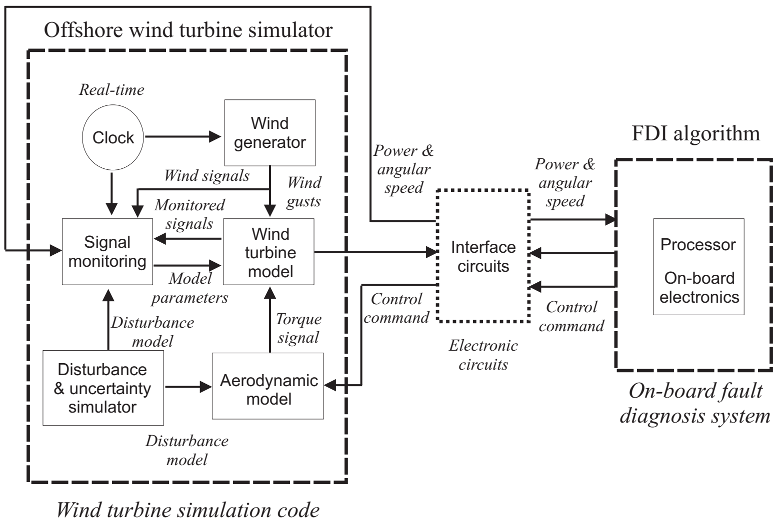

Figure 8. The block diagram of the offshore wind turbine HIL test–rig.

The set–up of the experimental test–rig, represented in Fig. 8, consists of three interconnected components:

- **Simulator**: the models of the offshore wind turbine system dynamics have been implemented in LabVIEW® environment, 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2018 doi:10.20944/preprints201812.0265.v1
Peer-reviewed version available at Appl. Sci. 2019, 9, 783; doi:10.3390/app9040783

- **On board electronics**: The fault diagnosis scheme 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 fault diagnosis algorithms, including the fault estimation modules of the resiudal generator banks.
- **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 system.

The evaluation of the performances of the considered fault diagnosis strategies in this more realistic scenario is based on the computation of the following indices [24]:

- **False Alarm Rate** (FAR): the ratio between the number of wrongly detected faults and the number of simulated faults;
- **Missed Fault Rate** (MFR): the ratio between the total number of missed faults and the number of simulated faults;
- **True FDI Rate** (TFR): the ratio between the number of correctly detected faults and the number of simulated faults (complementary to MFR);
- **Mean FDI Delay** (MFD): the delay time between the fault occurrence and the fault detection.

A suitable number of experiments has been performed in order to compute these indices and to test the robustness of the considered fault diagnosis schemes. Indeed, this experimental set–up is useful at these stage, as the efficacy of the diagnosis depends on both the model approximation capabilities, the model–reality mismatch, and the measurements errors.

Table 9 refers to the fuzzy fault diagnosis scheme and summarises the results obtained using this real–time HIL set–up for the offshore wind turbine system.

Table 9. 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

On the other hand, Table 10 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.

Table 10. 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

It is worth observing the effectiveness of the achieved results with this real–time test rig. However, 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 lead to further uncertainty effects. Note also 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 performances are interesting and the developed fault diagnosis systems can be also effectively considered for application to real offshore wind turbine installations.

5. Conclusion

The paper analysed the development of tools for solving the problem of the fault diagnosis of a wind turbine system. The design of this indicator relies on the direct estimate of the fault itself, which used two data–driven schemes. They are proposed as represented viable tools for coping with poor knowledge of the process dynamics, in presence of noise and disturbance effects. These data–driven schemes were based on fuzzy and neural network structures used to describe the nonlinear dynamic links between input–output measurements and the considered fault signals. The selected prototypes belong to the nonlinear autoregressive with exogenous input architectures, as they can describe any nonlinear dynamic relationship with arbitrary degree of accuracy. The fault diagnosis strategies were tested via a high–fidelity simulator describing the normal and the faulty behaviours of a wind turbine process. The achieved performances, in terms also of reliability and robustness, were thus verified by considering also the presence of uncertainty and disturbance effects included in the wind turbine simulator. Further works will consider the performance these fault diagnosis schemes when applied to real plants.

Sample Availability: The software simulation codes for the proposed fault diagnosis strategies and the proposed results are available from the authors in the Matlab and Simulink environments.

Acknowledgments: The research works have been supported by the FAR2018 local fund from the University of Ferrara. On the other hand, the costs to publish in open access have been covered by the FIR2018 local fund from the University of Ferrara.

Author Contributions: Silvio Simani conceived and designed the simulations; moreover, he analysed the methodologies and the achieved results; together with Paolo Castaldi, wrote the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Bibliography

1. Odgaard, P.F. FDI/FTC wind turbine benchmark modelling. Workshop on Sustainable Control of Offshore Wind Turbines; Patton, R.J., Ed.; Centre for Adaptive Science & Sustainability, , 2012; Vol. 1.
2. Lan, J.; Patton, R.J.; Zhu, X. Fault–tolerant wind turbine pitch control using adaptive sliding mode estimation. *Renewable Energy* **2018**, *116*, 219–231. DOI: 10.1016/j.renene.2016.12.005.
3. Habibi, H.; Nohooji, H.R.; Howard, I. Adaptive PID Control of Wind Turbines for Power Regulation with Unknown Control Direction and Actuator Faults. *IEEE Access* **2018**, *6*, 37464–37479. DOI: 10.1109/ACCESS.2018.2853090.
4. Odgaard, P.F.; Stoustrup, J. A Benchmark Evaluation of Fault Tolerant Wind Turbine Control Concepts. *IEEE Transactions on Control Systems Technology* **2015**, *23*, 1221–1228.
5. Parker, M.A.; Chong, H.N.; Ran, L. Fault–Tolerant Control for a Modular Generator–Converter Scheme for Direct–Drive Wind Turbines. *IEEE Trans. on Industrial Electronics* **2011**, *58*, 305–315.
6. Odgaaard, P.F.; Shafiei, S.E. Evaluation of wind farm controller based fault detection and isolation. Proceedings of the IFAC SAFEPROCESS Symposium 2015; Elsevier., Ed.; IFAC, Elsevier: Paris, France, 2015; Vol. 48, pp. 1084–1089. DOI: 10.1016/j.ifacol.2015.09.671.
7. Byrski, J.; Byrski, W. A double window state observer for detection and isolation of abrupt changes in parameters. *International Journal of Applied Mathematics and Computer Science* **2016**, *3*, 585–602. DOI: 10.1515/amcs-2016-0041.

8. Xu, F.; Puig, V.; Ocampo-Martinez, C.; Olaru, S.; Niculescu, S.I. Robust MPC for actuator–fault tolerance using set–based passive fault detection and active fault isolation. *International Journal of Applied Mathematics and Computer Science* **2017**, *27*, 43–61. DOI: 10.1515/amcs-2017-0004.
9. Babuška, R. *Fuzzy Modeling for Control*; Kluwer Academic Publishers: Boston, USA, 1998.
10. Simani, S.; Fantuzzi, C.; Rovatti, R.; Beghelli, S. Parameter Identification for Piecewise Linear Fuzzy Models in Noisy Environment. *International Journal of Approximate Reasoning* **1999**, *1*, 149–167. Publisher: Elsevier.
11. Roy, N.; Ganguli, R. Filter design using radial basis function neural network and genetic algorithm for improved operational health monitoring. *Applied Soft Computing Journal* **2006**, *6*, 154–169.
12. Odgaard, P.F.; Stoustrup, J.; Kinnaert, M. Fault–Tolerant Control of Wind Turbines: A Benchmark Model. *IEEE Transactions on Control Systems Technology* **2013**, *21*, 1168–1182. ISSN: 1063–6536. DOI: 10.1109/TCST.2013.2259235.
13. Chen, J.; Patton, R.J. *Robust Model–Based Fault Diagnosis for Dynamic Systems*; Kluwer Academic Publishers: Boston, MA, USA, 1999.
14. Takagi, T.; Sugeno, M. Fuzzy Identification of Systems and Its Application to Modeling and Control. *IEEE Transaction on System, Man and Cybernetics* **1985**, *SMC-15*, 116–132.
15. Jun, W.; Shitong, W.; Chung, F.L. Positive and negative fuzzy rule system, extreme learning machine and image classification. *International Journal of Machine Learning and Cybernetics* **2011**, *2*, 261–271. DOI: 10.1007/s13042–011–0024–1.
16. Graaff, A.J.; Engelbrecht, A.P. Clustering data in stationary environments with a local network neighbourhood artificial immune system. *International Journal of Machine Learning and Cybernetics* **2012**, *3*, 1–26. DOI: 10.1007/s13042–011–0041–0.
17. Fantuzzi, C.; Simani, S.; Beghelli, S.; Rovatti, R. Identification of piecewise affine models in noisy environment. *International Journal of Control* **2002**, *75*, 1472–1485. Publisher: Taylor and Francis, Ltd. DOI: 10.1109/87.865858.
18. Haykin, S. *Kalman Filtering and Neural Networks*; Adaptive and Learning Systems for Signal Processing, Communications, and Control, Wiley–Interscience: John Wiley & Sons, Inc. New York, USA, 2001.
19. Xu, J.X.; Liu, C.; Hang, C. Combined Adaptive and Fuzzy Control Using Multiple Models. Proc. third IEEE Int. Conf. on Fuzzy Syst.; , 1994.
20. Hunt, K.; Sbarbaro, D.; Zbikowki, R.; Gawthrop, P. Neural networks for control system: a survey. *IEEE Trans. Neural Networks* **1992**, *28*, 1083–1112.
21. Simani, S.; Farsoni, S.; Castaldi, P. Fault Diagnosis of a Wind Turbine Benchmark via Identified Fuzzy Models. *IEEE Transactions on Industrial Electronics* **2015**, *62*, 3775–3782. Invited paper for the special issue "Real–time fault diagnosis and fault tolerant control". DOI: 10.1109/TIE.2014.2364548.
22. Ding, S.X. *Model–based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, 1st ed.; Springer: Berlin Heidelberg, 2008. ISBN: 978–3540763031.
23. Simani, S. Application of a Data–Driven Fuzzy Control Design to a Wind Turbine Benchmark Model. *Advances in Fuzzy Systems* **2012**, *2012*, 1–12. Invited paper for the special issue: Fuzzy Logic Applications in Control Theory and Systems Biology (FLACE) . ISSN: 1687–7101, e–ISSN: 1687-711X. DOI: 10.1155/2012/504368.
24. Bartys, M.; Patton, R.; Syfert, M.; de las Heras, S.; Quevedo, J. Introduction to the DAMADICS Actuator FDI Benchmark Study. *Control Engineering Practice* **2006**, *14*, 577–596. Special Issue "Fault Diagnosis of Actuator Systems: the DAMADICS Benchmark Problem".