Article

Data–Driven Fault Diagnosis of a Wind Farm Benchmark Model

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

[1] Dipartimento di Ingegneria, Università degli Studi di Ferrara. Via Saragat 1E, Ferrara (FE) 44122, Italy; {silvio.simani,saverio.farsoni}@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

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

Abstract: The fault diagnosis of wind farms has been proven to be a challenging task and motivates the research activities carried out through this work. Therefore, this paper deals with the fault diagnosis of a wind park benchmark model, and it considers viable solutions to the problem of earlier fault detection and isolation. The design of the fault indicator involves data–driven approaches, as they can represent effective tools for coping with poor analytical knowledge of the system dynamics, noise, uncertainty and disturbances. In particular, the proposed data–driven solutions rely on fuzzy models and neural networks that are used to describe the strongly nonlinear relationships between measurement and faults. The chosen architectures rely on nonlinear autoregressive with exogenous input models, 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 farm installation. The achieved performances are also compared with those of a model–based approach relying on nonlinear differential geometry tools. 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 logic; neural networks; data-driven approaches; nonlinear geometric approach; wind farm benchmark simulator

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 generated the desired value of electrical power continuously, depending on the actual wind speed level and on the grid 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 failures. Wind turbines in the megawatt size 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 the 30%, particularly considering the offshore installation [1].

These considerations motivate the introduction of fault diagnosis systems that can be also coupled with fault tolerant controllers [2,3]. Actually, 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 failures that jeopardise wind turbine components, thus avoiding unplanned replacement of functional parts, as well as the reduction of the O & M costs and the increment of the energy production. The advent of computerised control, communication networks and information techniques

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

brings interesting challenges concerning the development of novel Fault Detection and Diagnosis (FDD) and Fault Tolerant Control (FTC) design strategies for industrial processes.

Indeed, in the recent years, many contributions have been proposed related to the topics of fault diagnosis of wind turbines and wind farms, see *e.g.* [4,5]. Some of them highlight the difficulties to achieve the diagnosis of particular faults, *e.g.* those affecting the drive–train, at wind turbine level. However these fault are better dealt with at wind farm level, when the wind turbine is considered in comparison to other wind turbine of the wind farm [2]. Moreover, fault tolerant control of wind turbines has been investigated *e.g.* in [3,6] and international competitions on these issues arose [7,8].

Therefore, the fault diagnosis of wind turbine and wind farm systems has been proven to be a challenging task and motivates the research activities carried out through this paper.

In recent years, the increasing demand for energy generation from renewable sources has led to a growing 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 control systems can also provide the optimisation of energy conversion and guarantee the desired performances even in presence of possible anomalous working condition, caused by unexpected faults and anomalous working conditions.

This work deals with the fault diagnosis of a wind farm system, and it proposes the application of viable and reliable solutions to the problem of earlier Fault Detection, Isolation (FDI) and estimation. Further fault tolerant controllers, which are not considered in this work, can be based on the fault diagnosis module developed in this paper, that provides the on–line information on the faulty or fault–free status of the system and the fault estimation, so that the controller action can be compensated. The design of the fault diagnosis system involves data–driven approaches, as they offer an effective tool for coping with a poor analytical knowledge of the system dynamics, noise, uncertainty and disturbance.

The first data–driven proposed solution relies on fuzzy Takagi–Sugeno models, which are derived from a clustering c–means algorithm, followed by an identification procedure solving the noise–rejection problem. Furthermore, 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 classic back–propagation Levenberg–Marquardt algorithm, that processes a set of acquired target data.

A purely nonlinear model–based scheme for fault tolerant control purpose was also proposed by the same authors, which is based on NonLinear differential Geometric Approach (NLGA) tools [9]. Already suggested by the authors also in the aerospace framework [10], it was extended by the same authors to the active fault tolerant control for the same wind farm simulator [11], but it is considered here only for comparison purpose.

The developed fault diagnosis schemes are tested by means of a high–fidelity benchmark model, that simulates the normal and the faulty behaviour of the wind farm system. The performances achieved via the data–driven approaches are analysed and compared with the proposed nonlinear model–based approach. Moreover, a Monte–Carlo analysis validates the robustness and reliability features of the proposed fault diagnosis schemes against the typical parameter uncertainties and disturbances. Finally, the effectiveness shown by the obtained results suggests further investigations on the industrial application of the proposed fault diagnosis methodologies.

The work is organised as follows. Section 2 recalls the wind farm benchmark simulator. Section 3 describes the FDD schemes for residual generation relying on fuzzy prototypes and neural network structures. A purely nonlinear approach proposed by the authors for wind turbine FDD is summarised in Section 4. The achieved results are reported in Section 5. Comparisons with the recalled model–based

nonlinear FDD strategy is also reported. Finally, Section 6 concludes the paper by summarising the main achievements of the work, and providing some suggestions for further research issues.

2. Wind Farm Benchmark Model

The wind farm benchmark model considered in this monograph has been proposed in [2] by the same authors who developed the wind turbine benchmark model [3].

It consists of 9 wind turbines arranged in a squared grid of 3 rows and 3 columns. The complete wind farm model consists of 3 main submodels: the wind and wake model, the plant model, and the controller model, interacting as sketched in Figure 1. The layout of the wind farm with 9 turbines of the square grid and the masts along the wind directions are sketched in Figure 2.

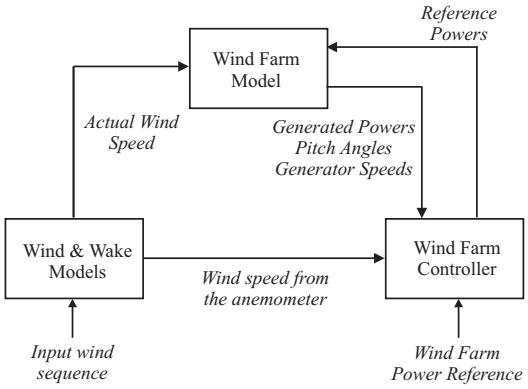

Figure 1. Block diagram of the wind farm benchmark model.

The distance between the wind turbines in both directions is 7 times the rotor diameter, L. Two measuring masts are located in front of the wind turbines, one in each of the wind directions considered in this benchmark model, *e.g.* 0 deg and 45 deg. The wind speed is measured by these measuring masts which are located in a distance of 10 times L in front of the wind farm. In this way, the measuring devices are not affected by the wind turbine wakes and are able to provide sufficiently accurate wind speed measurements. The wind turbines of the farm are defined by their row and column indices in the coordinate system, as illustrated in Figure 2.

The farm uses generic 4.8 MW wind turbines, which are three–bladed horizontal–axis, pitch–controlled variable–speed wind turbines. Each of the wind turbines is described by simplified models including control logics, variable parameters and 3 states [2].

2.1. Wind and Wake Model

The wind and wake model provides the wind speed for each of the 9 turbines, contained in the vector $\mathbf{v}_w$, as well as for a measuring mast $v_{w,m}$. They are determined starting from a certain wind sequence (two different wind sequences are included in the simulator) and their elaboration takes into account the delay and the interaction among the turbines depending on wind direction. In particular, the wake effect is described as reported in [12] by means of a static deficit coefficient of 0.9. Finally the turbulence is modelled by an additive Gaussian white noise.

2.2. Wind Farm Benchmark Overall Model

The plant model represents the 9 wind turbines with the same submodel for each of them. It receives as input the $\mathbf{v}_w$ vector and the $\mathbf{P}_r$ vector containing the 9 reference signals from the controller. The outputs are the vectors $\mathbf{P}_g$, $\boldsymbol{\beta}$, $\boldsymbol{\omega}_g$ that contain the generated powers, the pitch angles, and the generator speeds, respectively, for each of the 9 turbines.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

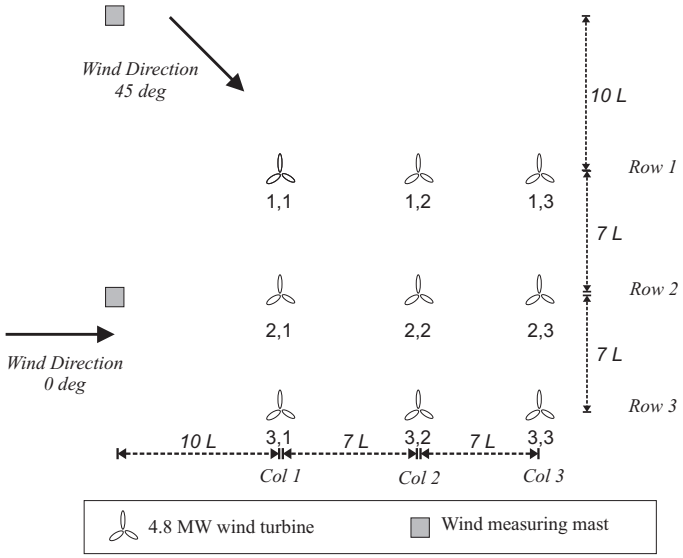

Figure 2. Layout of the wind farm benchmark with 9 wind turbines.

Inside the turbine submodel, the current wind speed is elaborated by means of a look–up table in order to compute the available power $P_w(t)$. Then, the generated power is computed according to the relation of Eq. (1):

$$P_g(t) = P_c(t) + \gamma_p \sin(2\pi\sigma_p t) \tag{1}$$

where the first term $P_c(t)$ is equal to the current lower value between the filtered available power $\hat{P_w}$ and the filtered reference power $\hat{P}_r$:

$$P_c(t) = \begin{cases} \hat{P_w}(t), & \text{if } \hat{P_w}(t) < \hat{P}_r(t) \\ \hat{P}_r(t), & \text{if } \hat{P_w}(t) > \hat{P}_r(t) \end{cases} \tag{2}$$

The second term of the Eq. (1) represents the oscillations caused by the drive–train, whose amplitude is γ_p and frequency σ_p.

The filtered signals $\hat{P_w}(t)$ and $\hat{P}_r(t)$ differs from the input variables by means of a first order transfer function in the form of Eqs. (3) and (4):

$$\hat{P_w}(s) = \frac{\tau_w(v_w)}{s + \alpha_w(v_w)} P_w(s) \tag{3}$$

$$\hat{P}_r(s) = \frac{\tau_p}{s + \tau_p} P_r(s) \tag{4}$$

where the parameter τ_p is a fixed value, while the functions $\tau_w(\cdot)$ and $\alpha_w(\cdot)$ depend on the wind speed and are computed by means of a look–up table. Note that descriptions using maps rather than analytical functions allow to provide a a computationally simple, but at the same time, also a realistic simulator of wind farm system. On the other hand, the relations of Eqs. (3) and (4) describe the filtering effect of the wind farm on the available and reference powers, respectively, which cannot change instantaneously.

The relation between the reference pitch signal and the actual pitch angle of the wind turbine of the wind farm has been described by a first–order transfer function in the form of Eq. (5):

$$\beta(s) = \frac{\tau_\beta}{s + \tau_\beta} \beta_r(s) \tag{5}$$

With respect to the second–order system already considered *e.g.* in [3], this means that the wind turbine blades of wind farm simulator are actuated by means of electric motors rather than hydraulic systems.

Then, the generator speed of each turbine is modelled as described in Eq. (6):

$$\omega_g(t) = f_\omega\left(P_c(t)\right)\left(1 + \frac{\gamma_p}{\omega_{g,\,max}}\,\sin(2\,\pi\,\sigma_p\,t)\right) \tag{6}$$

where $f_\omega(\cdot)$ is computed by means of a look–up table, whilst the oscillation term, due to the drive–train, has an amplitude equal to the ratio between the parameter γ_p and the maximal generator speed $\omega_{g,\,max}$. In this way, the relation of Eq. (6) allows to include both the generator controller and the effect of the drive–train oscillations, which depend on its time–constant $1/\sigma_p$ and the damping factor γ_p. This term is also reduced by the maximal generator speed $\omega_{g,\,max}$.

Finally, the wind farm controller forces each turbine to follow a reference power signal $P_r(k)$ that is $1/9$ of the wind farm power reference. Moreover, in order to avoid fast variation of the control signal, the wind farm power reference is low–pass filtered obtaining $\hat{P}_{wf,r}$. The controller is modelled as discrete–time system and uses a sample frequency of 10 Hz [2]:

$$P_r(k) = \frac{1}{9}\,\hat{P}_{wf,r}(k) \tag{7}$$

2.3. Wind Farm Fault Scenario and Model Parameters

3 common fault scenarios are considered [2]. They affect the output variables of the plant system. These kinds of fault are difficult to detect considering the single wind turbine, but they can be identified at wind farm level.

The first fault considered represents the debris build–up on the blade surface. Its effect corresponds to a change of the aerodynamic of the affected turbine and the consequent decreasing of the generated power that is modelled by a scaling factor of 0.97 applied to the generated power signal. An analysis of this kind of fault is reported in [13].

The second fault is a misalignment of one blade caused by an imperfect installation. The effect is an offset between the actual and the measured pitch angle of the affected turbine. This fault can excite structural modes and creates undesired vibrations that can damage the system severely. The faulty signal involves an offset of 0.3 deg on the pitch angle.

The third fault represents the wear and tear in the drive–train. It was demonstrated [7] that this kind of fault is difficult to detect at wind turbine level, and the current trend is to analyse the frequency spectra of different vibration measurement. In this benchmark model the fault affects the generated power increasing the amplitude of its oscillation of the 26% of the nominal value, and the generator speed increasing the amplitude of its oscillation of the 130%.

Table 1 summarises the considered faults, with a brief description of their typology and topology.

Table 1. The fault scenarios considered for the wind farm benchmark model.

Fault case	Fault description	Fault model	Fault effect
1	Blade surface debris build–up	Scaling factor (aerodyn. model)	Reduced generated power
2	Blade misalignment	Sensor offset	Measured pitch angle offset
3	Drive–train wear and tear	Drive–train model params.	Generator speed oscillation

Finally, Table 2 reports the parameters used in the wind farm benchmark model.

3. Data–Driven Fault Diagnosis Strategies

This section introduces the fault diagnosis schemes adopted in the simulations summarised in Section 5. Firstly, the fault diagnosis strategy is described in Section 3.1, followed by the summary of the design of the suggested data–driven residual generation methodologies relying on fuzzy prototypes and neural networks recalled in Sections 3.3 and 3.4, respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

Table 2. Parameter values of the wind farm benchmark model.

Variable name	Parameter value
R	57.5m
τ_p	$1.2 rad/s$
γ_p	1000W
σ_p	10Hz
τ_β	$1.6 rad/s$
$\omega_{g,max}$	$158 rad/s$

3.1. Fault Diagnosis Scheme

In the following the discrete–time monitored systems, *i.e.* the faults on the input and output measurements that are acquired from the wind farm simulator, as represented in Figure 3, are described in the form of Eq. (8):

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

where $\mathbf{u}^*(k)$ and $\mathbf{y}^*(k)$ are the actual unmeasurable variables, whilst $\mathbf{u}(k)$ and $\mathbf{y}(k)$ represent the sensor measurements, affected by both the measurement noise and the faults. $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$ are additive fault signals, that assume values different from zero only in presence of faults. Note that these terms represent *equivalent* fault signals that produce the same effects of the actual faults affecting the wind farm, as described in Section 2.3.

The same considerations hold for the measurement noise or error signals $\tilde{\mathbf{u}}(k)$ and $\tilde{\mathbf{u}}(k)$. They are modelled as white Gaussian process additive noise terms on the actual measurements acquired from the process under investigation, and are equivalent signals representing the actual effect of the measurement errors. This representation is also known as Errors–In–Variables (EIV) model [14].

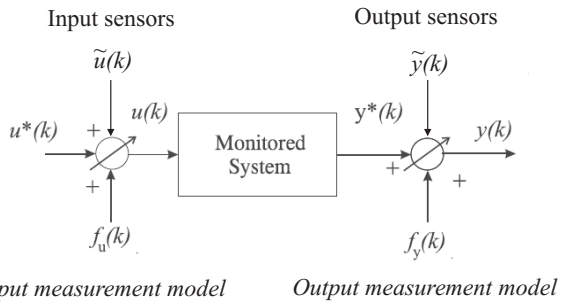

Figure 3. The equivalent fault and measurement error description of the system under diagnosis.

Figure 3 shows the general scheme with the faults affecting the system under diagnosis, *i.e.* the wind farm, as equivalent additive signals on the input and output measurements (sensors).

Among the different approaches to generate the residual signals available in the related literature, the solution proposed in this work exploits two approaches. They rely on fuzzy and neural network models, which provide on–line estimations of the fault signals $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$. Hence, as shown in Figure 4, the residuals $\mathbf{r}$ are represented by the estimated fault signals $\hat{\mathbf{f}}(k)$ generated by the fault estimators filtering the measured (sampled) inputs and outputs $\mathbf{u}(k)$ and $\mathbf{y}(k)$, respectively:

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

The generic i–th signal $f_i(k)$ of the vector $\mathbf{f}(k)$ represents one of the components of the faults $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$ affecting one of the input or output measurement sensors. This equivalent signal representation is used to model the actual faults of the wind farm described in Section 2.3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

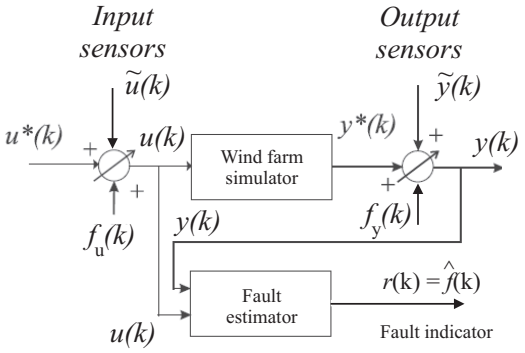

Figure 4. The general residual generation scheme relying on fault estimators.

Figure 4 sketches the residual generation scheme that is achieved via the fault estimator system, by using the acquired input and output measurements $\mathbf{u}(k)$ and $\mathbf{y}(k)$. The fault diagnosis process requires, as first step, the FDI tasks. 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 any evaluation functions, as addressed *e.g.* in [15]:

Therefore, the occurrence of the i–th fault can be simply detected via the threshold logic of Eqs. (10) applied to the i–th residual $r_i(k)$:

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

where the i–th component $r_i(k)$ of the residual vector $\mathbf{r}(k)$ of Eq. (9) is considered a random variable, whose unknown mean $\bar{r}_i$ and variance $\sigma^2_{r_i}$ are estimated in fault–free condition, after the acquisition of N samples, as shown by the relations of Eqs. (11):

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

The tolerance parameter δ has to be properly tuned in order to separate the fault–free from the faulty situations. The δ value 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 the δ value.

Consequently to the fault detection, the fault isolation task is easily achieved by means of a bank of estimators. As described by Eqs. (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 these assumptions, 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 this estimators is equal to the number of faults that have to be diagnosed, *i.e.* in general equal to the number of input and output measurements, $r + m$. Therefore, in general the i–th fault estimator that reconstructs the fault $\hat{f}(k) = r_i(k)$ should be 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 3.2. For each fault case, the failure modes and their resulting effects on the rest of the system are analysed, 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 Figure 5.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

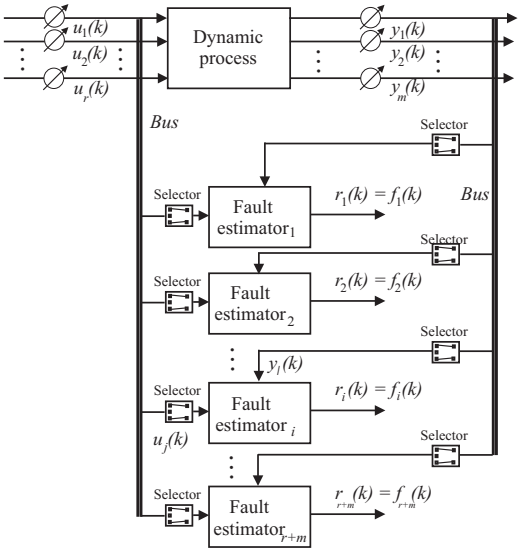

Figure 5. The general estimator scheme for the reconstruction of the *equivalent* input or output faults, $\mathbf{f}_u$ or $\mathbf{f}_u$.

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 addressed in Section 3.2, 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 could be correctly isolated using this configuration, if the FMEA procedure is properly performed.

The capabilities of the adopted estimator banks can be summarised by means of the so–called *fault signature* matrix, depicted in Table 3, where each entry that is characterised by a value equal to '1' means that the considered residual (*i.e.* the equivalent fault) is sensitive to the actual fault effect ('0' otherwise), under the hypothesis above mentioned.

Table 3. Fault signatures for the generalised fault estimator bank.

	u_1	u_2	...	u_r	y_1	y_2	...	y_r
r_1	1	0	...	0	0	0	...	0
r_2	0	1	...	0	0	0	...	0
⋮			⋱				...	⋮
r_r	0	0	...	1	0	0	...	0
r_{r+1}	0	0	...	0	1	0	...	0
r_{r+2}	0	0	...	0	0	1	...	0
⋮			...				⋱	⋮
r_{r+m}	0	0	...	0	0	0	...	1

As already remarked, the FMEA tool [16], that 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, that is briefly remarked here, will be addressed in more detail in Section 3.2. Then, the design of the fuzzy or neural network models can be performed, as recalled in Sections 3.3 and 3.4, respectively. Finally, the threshold test logic of Eq. (10) allows the achievement of the fault diagnosis tasks.

3.2. Failure Mode & Effect Analysis for the Wind Farm System

By following the guidelines reported in [16], the FMEA procedure has been performed on the wind farm simulator. The FMEA is a sensitivity analysis aimed at estimating the most sensitive

measurements with respect to the simulated fault conditions. 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 (RMSE) between the fault–free and faulty measured signals are computed, so that, for each fault $f_i(k)$, the most sensitive signals ($u_j(k)$ and $y_l(k)$) to that specific fault are selected.

In particular, the FMEA is conducted on the basis of a selection algorithm that is achieved by introducing the normalised sensitivity function N_x, defined in the form of by Eq. (12):

$$N_x = \frac{S_x}{S_x^*} \tag{12}$$

where:

$$S_x = \frac{\left\| x_f(k) - x_n(k) \right\|_2}{\| x_n(k) \|_2} \tag{13}$$

$$S_x^* = \max \frac{\left\| x_f(k) - x_n(k) \right\|_2}{\| x_n(k) \|_2} \tag{14}$$

Its value represents the effect of the considered fault case with respect to a certain measured signal $x(k)$. The subscripts 'f' and 'n' indicate the faulty and the fault–free case, respectively. Therefore the measurements mostly affected by the considered fault imply a value of N_x equal to '1'. Otherwise, smaller values of N_x mean that the signal x is not practically affected by the fault. The signals characterised by the highest value of N_x can be selected as the most sensitive measurements and they will be considered in the design of the fault diagnosis blocks.

The results of the FMEA sensitivity are reported in Table 4 for the wind farm benchmark. The selected signals for each fault included in the wind farm benchmark are divided as inputs and outputs.

Table 4. FMEA and selected benchmark signals.

Fault case $f_i(k)$	**Most Sensitive Measurements** $u_j(k)$ and $y_l(k)$
1	β_2, $P_{g,2}$, β_7, $P_{g,7}$, $v_{w,m}$
2	β_1, $\omega_{g,1}$, β_5, $\omega_{g,5}$, $v_{w,m}$
3	β_6, $P_{g,6}$, β_8, $\omega_{g,8}$, $v_{w,m}$

As a result, the fault diagnosis blocks that are designed as shown in Figure 5 exploit the reduced subset of input and output signals $u_j(k)$ and $y_l(k)$ of Table 4, instead of using all the system measurements recalled in Section 2.2. Moreover, this leads to a noteworthy simplification of the residual generator structure, thus providing also a decrease in the computational effort.

Finally, the threshold test logic of Eq. (10) allows the achievement of the fault diagnosis task. The design of the fuzzy systems and neural networks will be recalled in the next Sections 3.3 and 3.4, respectively.

3.3. Fuzzy System Modelling

This section describes the design of the fault estimators that is achieved by means of Takagi–Sugeno (TS) prototypes [17]. Indeed, the unknown relationships between the measurements acquired from the wind farm simulator and the considered faults are described by fuzzy models. They consist of a number of rules connecting the inputs with the output of the system under diagnosis, on the basis of a knowledge of its dynamics in form of IF $\Longrightarrow$ THEN relations, processed by fuzzy reasoning [18].

The approximation of nonlinear Multi–Input Single–Output (MISO) systems (but also extension to MIMO systems can be considered) is achieved by the TS fuzzy reasoning, as reported *e.g.* in [19,20].

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

According to the TS modelling approach, the consequents become crisp functions of the input, while the antecedents remain fuzzy propositions, therefore the fuzzy rule takes the form of Eq. (15) [17]:

$$R_i: \quad IF \quad (\text{fuzzy combination of inputs}) \quad THEN \quad \text{output} = g_i(\text{inputs}) \tag{15}$$

where i indicates the number of rules, with $i = 1, \ldots, n_C$. 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 functions g_i have a defined structure: it is the instance of parametrised function in affine form:

$$h_i = \mathbf{a}_i^T \mathbf{x} + b_i \tag{16}$$

where $\mathbf{a}_i$ is a parameter vector and b_i is a scalar offset, while h_i 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 local affine relations can be assumed [18]. 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 local affine models.

Thus, the TS inference takes the form of the simple algebraic expression of Eq. (17):

$$\hat{f} = \frac{\sum_{i=1}^{n_C} \lambda_i(\mathbf{x})\, h_i}{\sum_{i=1}^{n_C} \lambda_i(\mathbf{x})} \tag{17}$$

According to this description, the estimated fault $\hat{f}$ is the weighted average of affine functions of the measured input and output signals, where the weights are the combined degree of fulfilment of the system input and output measurements $\mathbf{u}(k)$ and $\mathbf{y}(k)$.

It is worth noting that the nonlinear system under diagnosis has a dynamic behaviour: therefore, the considered model input vector $\mathbf{x}$ can contains the current as well as the previous samples of the process input and output measurements $\mathbf{u}(k)$ and $\mathbf{y}(k)$.

Indeed, in order to introduce the time dependence into the model of Eq. (15), 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 Eq. (18):

$$\begin{aligned} \mathbf{x}(k) \quad = \quad & \Big[y_{j_1}(k-1), \ldots, y_{j_1}(k-o), \ldots, y_{j_l}(k-1), \ldots, y_{j_l}(k-o), \\ & u_{h_1}(k), \ldots, u_{h_1}(k-o), \ldots, u_{h_p}(k), \ldots, u_{h_p}(k-o) \Big]^T \end{aligned} \tag{18}$$

where u_{h_p} and y_{j_l} are the actual system input and output vectors, whose subscript indices j_l and h_p depend on the FMEA selection procedure of Section 3.2, with $1 \leq j_l \leq m$ and $1 \leq h_p \leq m$. k is the time step, with $k = 1, 2, \ldots, N$. The affine parameters of Eq. (16) can be grouped as in the expression of Eq. (19):

$$\mathbf{a}_i = \left[\alpha^{(i)}_{1,j_1}, \ldots, \alpha^{(i)}_{o,j_1}, \ldots, \alpha^{(i)}_{1,j_l}, \ldots, \alpha^{(i)}_{o,j_l}, \delta^{(i)}_{1,h_1}, \ldots, \delta^{(i)}_{o,h_1}, \cdots, \delta^{(i)}_{1,h_p}, \ldots, \delta^{(i)}_{o,h_p} \right]^T \tag{19}$$

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

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

Indeed, fuzzy clustering allows an item to belong to several cluster simultaneously, with different degrees of fulfilment, whereas the classic crisp clustering relies on mutual exclusive subsets. Different clustering methods have been proposed in literature, see *e.g.* the review [21], or more recent works [22,23].

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

$$\mathbf{Z} = \begin{bmatrix} z_{11} & \dots & z_{1N} \\ \vdots & \ddots & \vdots \\ z_{n1} & \dots & z_{nN} \end{bmatrix} \tag{20}$$

where n is the data dimension, and 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 $\mathbf{Z}$ of Eq. (20) is defined;
- the so–called fuzzy partition matrix $\mathbf{U} = [\mu_{ik}]$ is defined, which contains the values of the membership function for the couple i–th measurement/k–th cluster;
- the vector $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_{n_C}]$ containing the cluster prototypes is defined; they have to be determined and represent the centres from which the distance of each measurement is computed.

The widespread c–means goal function adopted in this work was formulated in [24] and defined in the form of Eq. (21):

$$J(\mathbf{Z}, \mathbf{U}, \mathbf{V}) = \sum_{i=1}^{n_C} \sum_{k=1}^{N} (\mu_{ik})^m D^2_{ik\mathbf{A}} \tag{21}$$

$m > 1$ being the weighting exponent, and:

$$D^2_{ik\,\mathbf{A}} = \|\mathbf{z}_k - \mathbf{v}_i\|^2_{\mathbf{A}} = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{z}_k - \mathbf{v}_i) \tag{22}$$

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

The minimisation algorithm exploits a series of Picard iterations that updates the cluster prototypes $\mathbf{V}$ and of the partition matrix $\mathbf{U}$, until a stop criterion is met [18].

An important point concerns the determination of the optimal number of clusters n_C, since the clustering algorithm assumes a suitable number of clusters, regardless of whether they are really present in the data or not. Once the partition matrix $\mathbf{U}$ has been estimated, the antecedent degrees of fulfilment μ_{ik} in $\mathbf{U}$ are easily estimated by interpolation or curve fitting methods [18].

3.3.1. Fuzzy Model Parameter Estimation

After the determination of $\mathbf{U}$ and $\mathbf{V}$, the design of the FIS can be considered as system identification problem since it requires the estimation of the consequent parameters $\mathbf{a}_i$ and b_i of Eq. (16) in a noisy environment. The identification scheme adopted in this work was proposed by the same authors in [25] and successfully exploited in the approximation of nonlinear functions through the piecewise affine models, as shown *e.g.* in [14]. This approach is based on the minimisation of the prediction errors of the individual TS local models considered as n_C independent problems. Moreover, their solutions rely on the so–called Frisch scheme [26] that is usually exploited in connection with the identification EIV models [14]. This assumption is motivated by the relations of Eqs. (8), where the input and output measurements are affected by both noise and fault effects.

In fact, the discrete–time dynamic model already represented in Figure 3 is considered in the EIV framework, the noise is supposed to affect the inputs $\mathbf{u}$ as well as the output measurements $\mathbf{y}$ as additive signals $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{y}}$ on the noise–free unmeasurable quantities $\mathbf{u}^*$ and $\mathbf{y}^*$, in the form of Eq. (23):

$$\begin{cases} \mathbf{u}(k) &= \mathbf{u}^*(k) + \tilde{\mathbf{u}}(k) \\ \mathbf{y}(k) &= \mathbf{y}^*(k) + \tilde{\mathbf{y}}(k) \end{cases} \tag{23}$$

Therefore, by considering the i–th TS consequent of the type of Eq. (17) and the associated local ARX model of order o with the regressors grouped into the vector $\mathbf{x}$ as in Eq. (18), the acquisition of N_i noisy measurement of input and output samples collected into the vector $\mathbf{x}$ permits the construction of the i–th data matrix $\mathbf{X}^{(i)}$ defined as:

$$\mathbf{X}^{(i)} = \begin{bmatrix} f(k) & \mathbf{x}^T(k) & 1 \\ f(k+1) & \mathbf{x}^T(k+1) & 1 \\ \vdots & \vdots & \vdots \\ f(k+N_i-1) & \mathbf{x}^T(k+N_i-1) & 1 \end{bmatrix} \tag{24}$$

with reference to the generic fault signal $f(k)$. The i–th covariance matrix $\mathbf{\Sigma}^{(i)}$ from the acquired data can be computed as:

$$\mathbf{\Sigma}^{(i)} = \mathbf{X}^{(i)^T}\mathbf{X}^{(i)} \geq 0 \tag{25}$$

that is a positive–definite matrix consisting in the sum of two terms:

$$\mathbf{\Sigma}^{(i)} = \mathbf{\Sigma}^{(i)^*} + \bar{\tilde{\mathbf{\Sigma}}}^{(i)} \tag{26}$$

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 $\bar{\tilde{\sigma}}_f$ and $\bar{\tilde{\sigma}}_{\mathbf{x}}$ through the expression:

$$\bar{\tilde{\mathbf{\Sigma}}}^{(i)} = \mathbf{diag}\left[\bar{\tilde{\sigma}}_f \mathbf{I}, \bar{\tilde{\sigma}}_{\mathbf{x}} \mathbf{I}, 0\right] \tag{27}$$

The solution of the identification problem above mentioned requires the estimation of $\bar{\tilde{\sigma}}_f$ and $\bar{\tilde{\sigma}}_{\mathbf{x}}$, that can be performed solving the expression in the form of Eq. (28):

$$\mathbf{\Sigma}^{(i)^*} = \mathbf{\Sigma}^{(i)} - \tilde{\mathbf{\Sigma}}^{(i)} \tag{28}$$

with:

$$\tilde{\mathbf{\Sigma}}^{(i)} = \mathbf{diag}\left[\tilde{\sigma}_f \mathbf{I}, \tilde{\sigma}_{\mathbf{x}} \mathbf{I}, 0\right] \tag{29}$$

in the unknown variables $\tilde{\sigma}_{\mathbf{x}}$ and $\tilde{\sigma}_f$.

In case all the assumption regarding the Frisch scheme [25] are satisfied, there exists one common point belonging to all the surfaces $\mathbf{\Gamma}^{(i)} = 0$ determined as the root locus of Eq. (28), that represents the actual noise variance values $\left(\bar{\tilde{\sigma}}_{\mathbf{x}}, \bar{\tilde{\sigma}}_f\right)$. 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. More details regarding this issue can be found *e.g.* in [14,26].

After the computation of the variances, the covariance noise matrix can be built as in Eq. (27), and the linear parameters in each cluster (therefore in each TS consequent) can be finally determined as a solution of the following expression:

$$\left(\mathbf{\Sigma}^{(i)} - \bar{\tilde{\mathbf{\Sigma}}}^{(i)}\right)\mathbf{a}_i = 0 \tag{30}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

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 for estimating the generic fault signal $f(k)$. In this section, after a brief introduction on the general structure, the properties and the functioning of a neural network, as well as its architecture are recalled. They will be exploited for implementing the neural network fault estimators of Figure 5.

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 [27] 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 the 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, denominated *activation function*, represents the engine of the neuron, as shown in Figure 6.

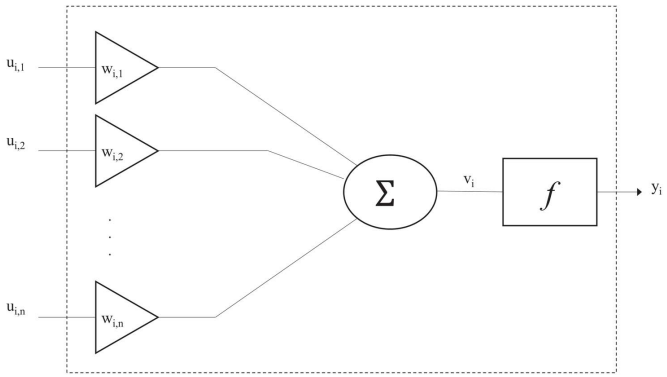

Figure 6. The i–th neuron model comprising the neural network.

A structural categorisation of neural networks concerns the way in which their elements are connected each others [28]. In a *feed–forward network*, also called *multilayer perceptron*, neurons are 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, that has to be equal to the number of actual network outputs. On the other hand, *recurrent networks* [29] 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.

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 model or predict unknown functions. In particular the open–loop NARX network belongs to this latter category as its inputs are delayed samples of the system inputs and outputs. Indeed, if properly trained, a NARX network can estimate the current (or the next) system output on the basis of the acquired past measurements of system inputs and outputs.

Generally speaking, considering a MISO system, the elaborations of the NARX neural network follow the law described by the relation of Eq. (31):

$$\begin{aligned} \hat{f}(k) \quad = \quad & f_{net}\Big(u_{h_1}(k-1), \ldots, u_{h_1}(k-d_u), \ldots, u_{h_p}(k-1), \ldots, u_{h_p}(k-d_u) \\ & y_{j_1}(k-1), \ldots, y_{j_1}(k-d_y), \ldots, y_{j_l}(k-1), \ldots, y_{j_l}(k-d_y)\Big) \end{aligned} \tag{31}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

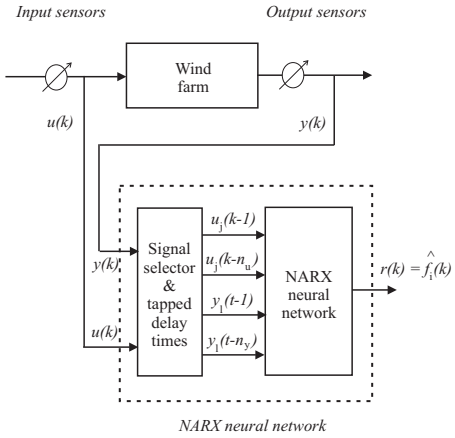

Figure 7. The NARX neural network used as reconstructor of the general fault signal $\hat{f}_i(k)$.

where $\hat{f}(k)$ is the estimation of the generic i–th fault, whilst $u_{h_p}(\cdot)$ and $y_{j_l}(\cdot)$ are the h_p–th and j_l–th measured system inputs and outputs, respectively, properly selected according to the FMEA procedure of Section 3.2, with $1 \leq j_l \leq m$ and $1 \leq h_p \leq m$. k is the time step, whilst d_u and d_y are the number of delays for the input and the output signals, respectively. f_{net} is the function realized by the network, that depends on the layer architecture, the number of neurons, their weights and their activation functions. The general fault estimation task of this NARX neural network used as fault estimator for fault diagnosis is depicted in Figure 7.

The design parameters of the overall architecture are represented by its number of neurons and the connections between layers, whilst the values of the weights inside each neuron are derived from the neural network training algorithm.

A NARX 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 of the inputs $u_j(\cdot)$ and $y_l(\cdot)$ (patterns), and the desired output $f_i(\cdot)$ (target) for the relative inputs. This training phase can be implemented in two different ways:

- **Incremental mode**: each couple input–target generates an updating of the network weights;
- **Batch mode**: all inputs and targets are applied to the network before the weights are updated.

Although the latter training mode requires more memory storage capability, with respect to the former one, it is characterised by a faster convergence and produces smaller errors, thus it will be considered in the following. The training phase objective is the minimisation of a performance function E, which depends on the weight vector $\mathbf{w}$.

Generally speaking, considering a number P of available example patterns consisting in the input–target pairs $(\mathbf{u}_p, \mathbf{t}_p)$, with $p = 1, \ldots, P$, defining $\hat{\mathbf{y}}_p$ the output generated by the network fed by $\mathbf{u}_p$, the p–th error vector can be expressed as:

$$\mathbf{e}_p = [\mathbf{t}_p - \hat{\mathbf{y}}_p] = [e_{p,1}, \ldots, e_{p,M}]^T \tag{32}$$

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

$$\bar{\mathbf{e}} = [e_{1,1}, \ldots, e_{1,M}, \ldots, e_{P,1}, \ldots, e_{P,M}]^T \tag{33}$$

Consequently, the performance function becomes:

$$E(\mathbf{w}) = \frac{1}{P} \sum_{p=1}^{P} (\mathbf{t}_i - \hat{\mathbf{y}}_i)^2 = \frac{1}{P} \sum_{p=1}^{P} \sum_{m=1}^{M} e_{p,m}^2 \tag{34}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

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

Standard numerical optimisation algorithms 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:

$$\mathbf{g} = \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \left[\frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_N}\right]^T \tag{35}$$

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \cdots & \frac{\partial^2 E}{\partial w_1 w_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_N w_1} & \cdots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} \tag{36}$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \cdots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,2}}{\partial w_1} & \cdots & \frac{\partial e_{1,2}}{\partial w_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_{P,M}}{\partial w_1} & \cdots & \frac{\partial e_{P,M}}{\partial w_N} \end{bmatrix} \tag{37}$$

The further iterations of these algorithms consist of the updating of the parameters and the computation of the new values of the performance function, until a stop criterion is met. The updating rules of the most common optimisation algorithms (*i.e.* the gradient descent, the Newton, the Gauss–Newton and the Levenberg–Marquardt algorithm) are reported in Table 5.

Table 5. Updating rule parameters.

Optimisation 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	$\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I}\right)^{-1} \mathbf{J}_k \bar{\mathbf{e}}_k$

Table 5 summarises the parameters of the updating rules of the most common optimisation algorithms, aimed at the minimisation of the performance function E. 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 do not meaningfully change. Otherwise, the Newton and the Gauss–Newton algorithms provide a 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.

The *Levenberg–Marquardt* algorithm, originally proposed in [30], 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 a fast and a stable convergence and it represents a suitable tool to train a neural network. Indeed, as shown in Section 5, 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 in [31], uses a technique denominated *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.

3.4.1. Fault Diagnosis Design Procedure

The complete design flow relying on both fuzzy systems and neural networks used as fault estimators for fault diagnosis is summarised in Figure 8.

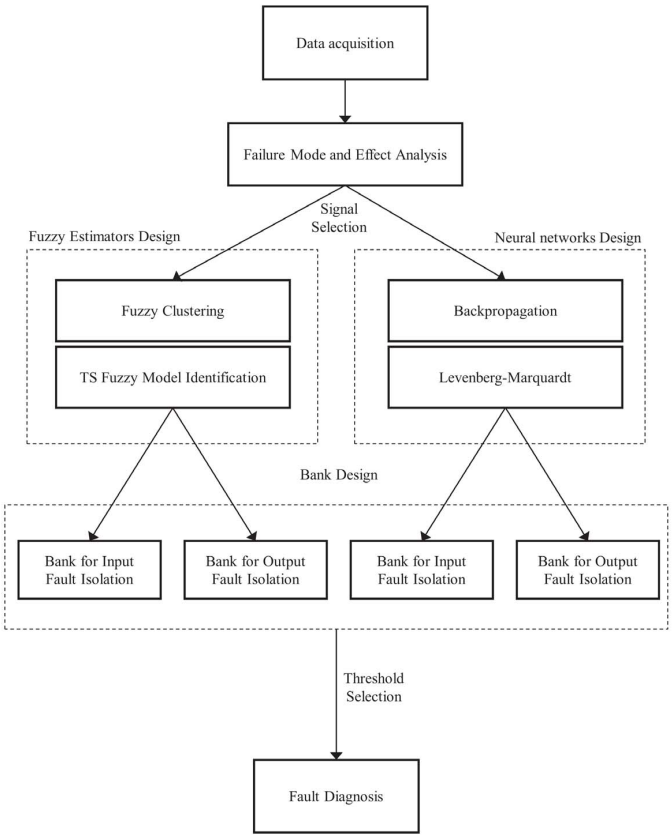

Figure 8. Fault diagnosis design procedure block diagram.

Finally, note that the design flow sketched in Figure 8 assumes that only single faults affect the process under diagnosis.

4. Model–Based Fault Diagnosis Method

This section recalls the purely nonlinear fault diagnosis scheme that represents a model–based approach mainly used for comparison purpose, as summarised in Section 5. In particular, this method is based on analytical physical law modelling and partially on parameter estimation of some coefficients describing this mathematical relations.

The proposed fault diagnosis scheme consists of three phases. The first step regards the estimation of the nonlinear disturbance distribution functions, which are required for the design of the nonlinear adaptive filter for fault estimation. The fault reconstruction is thus exploited for fault diagnosis purposes of the measured signals.

In order to achieve a robust and effective fault diagnosis solution, the fault estimations have to be decoupled from the disturbances acting on the system. Section 2 highlighted that these disturbances are represented by two effects. The first one derives from the wind signal affecting the i–th wind turbine model of the wind farm through its power coefficient C_p [2]. The decoupling of this effect was

already investigated in [11] but applied to a single wind turbine. It will be used here and applied to the wind turbine models of the wind farm. The second disturbance effect is due to the interactions among the wind turbines of the wind farm, and represented by the wind wakes [2].

Solutions dealing with the first disturbance term were based on the estimation of both the wind turbine power coefficient C_p values and the wind speed $v_w(t)$. It is worth noting that, regarding the wind wakes, a novel strategy based on a nonlinear design scheme is proposed here. In particular, as for the decoupling of the wind speed $v_w(t)$ addressed in [11], this approach required the analytical knowledge of the nonlinear disturbance distribution relation of the unknown inputs represented by the wake effects. In more detail, as shown in [11], the power coefficient C_p–map appearing in the wind turbine aerodynamic models of the wind farm was estimated by means of a two–dimensional polynomial representation, which was a function of the tip–speed ratio λ and the blade pitch angles β [32].

Once the disturbance description has been obtained in analytical form, the second stage of the fault diagnosis system design is based on the development of the nonlinear fault diagnosis filters. Their structure is obtained by exploiting a disturbance decoupling scheme belonging to the NLGA framework [33]. A coordinate transformation, highlighting a subsystem affected by the fault and decoupled by the disturbances, represents the starting point to design adaptive filters for fault estimation. It is worth observing that, by means of this NLGA approach, the fault estimate is decoupled from the disturbance d, which in this work are represented by the wake effects of the wind turbines affecting the i–th wind turbine of the wind farm.

Therefore, the proposed approach is applied to the general nonlinear model in the form of Eq. (38):

$$\begin{cases} \dot{x} &= n(x) + g(x)\, c + \ell(x)\, f + p_d(x)\, d \\ y &= h(x) \end{cases} \tag{38}$$

where the state vector $x \in \mathcal{X}$ (an open subset of $\mathbb{R}^{\ell_n}$), $c(t) \in \mathbb{R}^{\ell_c}$ is the control input vector, $f(t) \in \mathbb{R}$ is the fault, $d(t) \in \mathbb{R}^{\ell_d}$ is the disturbance vector, and $y \in \mathbb{R}^{\ell_m}$ is the output vector. $n(x)$, $\ell(x)$, the columns of $g(x)$, and $p_d(x)$ are smooth vector fields, with $h(x)$ a smooth map.

The development of the NLGA strategy for the design of the estimator for the fault f with the decoupling of the disturbance d is based on the procedure presented in [34]. It was shown that the considered NLGA scheme extended to the fault diagnosis problem is based on a coordinate change in the state space and in the output space, such that, by using the new (local) state and output coordinates $(\bar{x}, \bar{y})$, the system of Eq. (38) is transformed into [34]:

$$\begin{cases} \dot{\bar{x}}_1 &= n_1(\bar{x}_1, \bar{x}_2) + g_1(\bar{x}_1, \bar{x}_2)\, c + \ell_1(\bar{x}_1, \bar{x}_2, \bar{x}_3)\, f \\ \dot{\bar{x}}_2 &= n_2(\bar{x}_1, \bar{x}_2, \bar{x}_3) + g_2(\bar{x}_1, \bar{x}_2, \bar{x}_3)\, c + \\ & \quad + \ell_2(\bar{x}_1, \bar{x}_2, \bar{x}_3)\, f + p_2(\bar{x}_1, \bar{x}_2, \bar{x}_3)\, d \\ \dot{\bar{x}}_3 &= n_3(\bar{x}_1, \bar{x}_2, \bar{x}_3) + g_3(\bar{x}_1, \bar{x}_2, \bar{x}_3)\, c + \\ & \quad + \ell_3(\bar{x}_1, \bar{x}_2, \bar{x}_3)\, f + p_3(\bar{x}_1, \bar{x}_2, \bar{x}_3)\, d \\ \bar{y}_1 &= h(\bar{x}_1) \\ \bar{y}_2 &= \bar{x}_2 \end{cases} \tag{39}$$

with $\ell_1(\bar{x}_1, \bar{x}_2, \bar{x}_3)$ not identically zero. As remarked in [34], this procedure yields to the observable $\bar{x}_1$ subsystem of Eq. (39) that, if it exists, is affected by the faults f, and not affected by disturbances d.

This transformation can be applied to the system of Eq. (38), if and only if some fault detectability conditions are satisfied [34]. The system of Eq. (38) in the new reference frame is decomposed into 3 subsystems of Eqs. (39), where the first one (the so–called $\bar{x}_1$–subsystem) is always decoupled from the disturbances d and affected by the faults f, described in the form of Eq. (40):

$$\begin{cases} \dot{\bar{x}}_1 &= n_1(\bar{x}_1, \bar{y}_2) + g_1(\bar{x}_1, \bar{y}_2)\, c + \ell_1(\bar{x}_1, \bar{y}_2, \bar{x}_3)\, f \\ \bar{y}_1 &= h(\bar{x}_1) \end{cases} \tag{40}$$

where, as the state $\bar{x}_2$ in Eq. (39) is assumed to be measured, the variable $\bar{x}_2$ in Eq. (40) is considered as independent input, and denoted with $\bar{y}_2$.

The proposed NLGA adaptive filter is based on the least–squares algorithm with forgetting factor described by the adaptation law of Eq. (41):

$$\begin{cases} \dot{P} &= \beta P - \frac{1}{N^2} P^2 \breve{M}_1^2, & P(0) = P_0 > 0 \\ \dot{\hat{f}} &= P \epsilon \breve{M}_1, & \hat{f}(0) = 0 \end{cases} \tag{41}$$

with Eq. (42) representing the output estimation, and the corresponding normalised estimation error:

$$\begin{cases} \hat{\bar{y}}_{1s} &= \breve{M}_1 \hat{f} + \breve{M}_2 + \lambda \breve{\bar{y}}_{1s} \\ \epsilon &= \frac{1}{N^2} (\bar{y}_{1s} - \hat{\bar{y}}_{1s}) \end{cases} \tag{42}$$

where all the involved variables of the adaptive filter are scalar. In particular, $\lambda > 0$ is a parameter related to the bandwidth of the filter, $\beta \geq 0$ is the forgetting factor, and $N^2 = 1 + \breve{M}_1^2$ is the normalisation factor of the least–squares algorithm. Moreover, the proposed adaptive filter adopts the signals $\breve{M}_1$, $\breve{M}_2$, $\breve{\bar{y}}_{1s}$ which are obtained by means of a low–pass filtering of the signals M_1, M_2, $\bar{y}_{1s}$ as follows:

$$\begin{cases} \dot{\breve{M}}_1 &= -\lambda \breve{M}_1 + M_1, & \breve{M}_1(0) = 0 \\ \dot{\breve{M}}_2 &= -\lambda \breve{M}_2 + M_2, & \breve{M}_2(0) = 0 \\ \dot{\breve{\bar{y}}}_{1s} &= -\lambda \breve{\bar{y}}_{1s} + \bar{y}_{1s}, & \breve{\bar{y}}_{1s}(0) = 0 \end{cases} \tag{43}$$

The considered adaptive filter is described by the systems of Eqs. (41), (42), and (43). It is worth noting that in [34] it was showed that this adaptive filter provides an estimation $\hat{f}(t)$ that asymptotically converges to the magnitude of the actual fault f.

This completes the design of the fault diagnosis methodology based on NLGA adaptive filters, whose results will be shown in Section 5.

5. Simulation Results

The following simulations refers to the wind farm benchmark model when the fault diagnosis schemes based on the fuzzy and neural network fault estimators are designed.

The mean wind sequence driving all the simulations covers the most common operative range from 5 m/s up to 15 m/s, with a peak value of about 23 m/s. The wind and wake submodel described in Section 2 processes this sequence in order to generate the actual wind speed signal for all the turbines of the wind farm and for both the measurement musts associated with the two wind scenarios provided by the benchmark model, as shown in Section 2, also taking into account the disturbances and the interaction among turbines. Figure 9 depicts the leading wind sequence.

The available data consist of 440000 samples of input–output measurements, acquired with a sampling rate of 100 Hz.

The 3 proposed faults described in Section 2 affect different wind turbines at different times, by influencing the measured variables $\mathbf{u}(k)$ and $\mathbf{y}(k)$, and in particular the signals β_i, $\omega_{g,i}$, $P_{g,i}$ related with the i–th wind turbine of the wind farm. These faults are difficult to detect at wind turbine level. However, they can be more easily detected at wind farm level, by comparing the performance of different wind turbines. Table 6 shows the affected turbines per fault, together with the occurrence time.

These faults are simulated for the 3 cases of Table 1 and under the assumptions described in Section 2.3.

5.1. Fuzzy Estimator Results

The fuzzy fault estimators described in Section 3.3 are designed by selecting a number of $n_C = 5$ clusters and $o = 3$ delays on the input and output regressors, related to the signals selected by the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

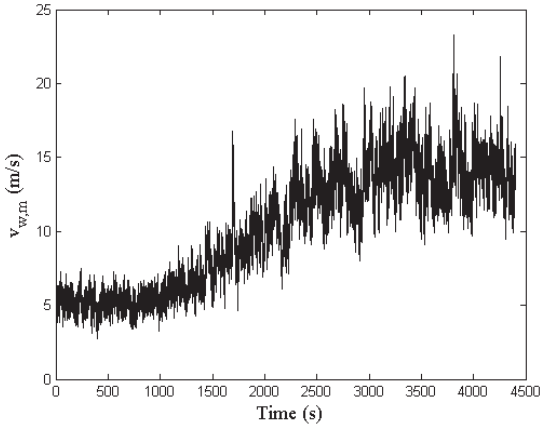

Figure 9. The mean wind speed sequence driving the simulations of the wind farm.

Table 6. Faulty turbines and activation times.

Fault	Affected Turbine	Time (s)
1	7	1000-1100
	2	3000-3100
2	1	1300-1400
	5	3300-3400
3	6	1600-1700
	8	3600-3700

FMEA recalled in Section 3.1. The TS models exploit Gaussian membership functions. Afterwards, the fault estimators are organised into a dedicated observer scheme of Section 3.1 that allows for the isolation of the 3 faults affecting the wind farm benchmark model described in Section 2.3.

Firstly, the capabilities of these 3 fuzzy fault estimators are evaluated in terms of RMSE, defined as the per cent difference between the measurements and the estimation, computed in fault free conditions. This index can be considered as the measure of the percentage of the data not correctly reconstructed by the estimator. As highlighted in Table 7, although this index increases when computed on data which are not used in the estimation phase (*i.e.* validation and test data sets), however, the percentage is always smaller than 5%, thus featuring a good modelling capability.

Table 7. RMSE of the fuzzy fault estimators.

Data Set	**RMSE** (%)		
	Fault 1 Estimator	Fault 2 Estimator	Fault 3 Estimator
Estimation	0.0090	0.0087	0.0092
Validation	0.0103	0.0101	0.0105
Test	0.0108	0.0103	0.0109

Then, the detection of the 3 faults is achieved by means of the residuals $r_i(k) = \hat{f}_i(k)$ generated by the fault estimators, after the proper tuning of the threshold parameter δ. In particular, Figure 10 highlights the residual $r_1(k)$ relative to the fault 1 estimator, whose value is bounded by the thresholds when the fault is not active, while it is significantly over the threshold when the fault occurs on the two different turbines. Similar results are achieved by the fault 2 and 3 estimators, whose residuals $r_2(k)$ and $r_3(k)$ are depicted in Figures 11 and 12, respectively.

5.2. Neural Network Estimator Results

In the same way of the fuzzy estimator approach, 3 NARX neural networks described in Section 3.4 have been designed to estimate the nonlinear behaviour between the measurements selected by

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

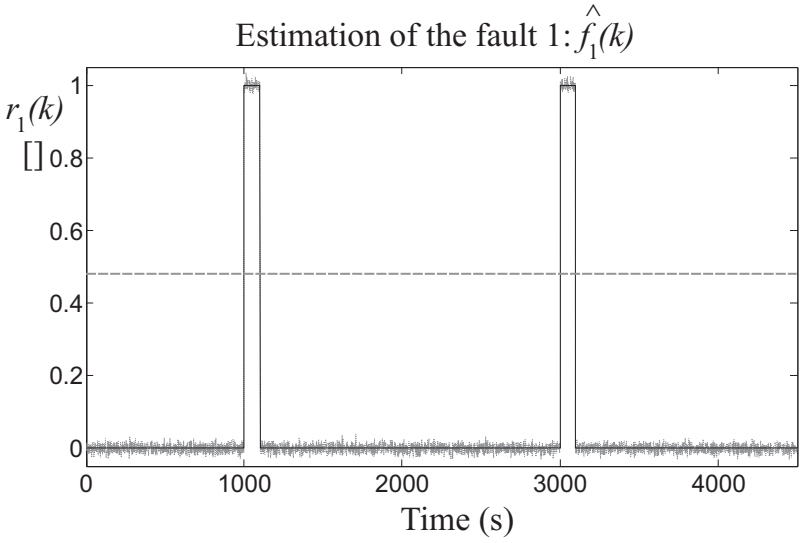

Figure 10. Fault 1 estimator residual $r_1(k)$ (continuous lines) and its threshold level (dotted line).

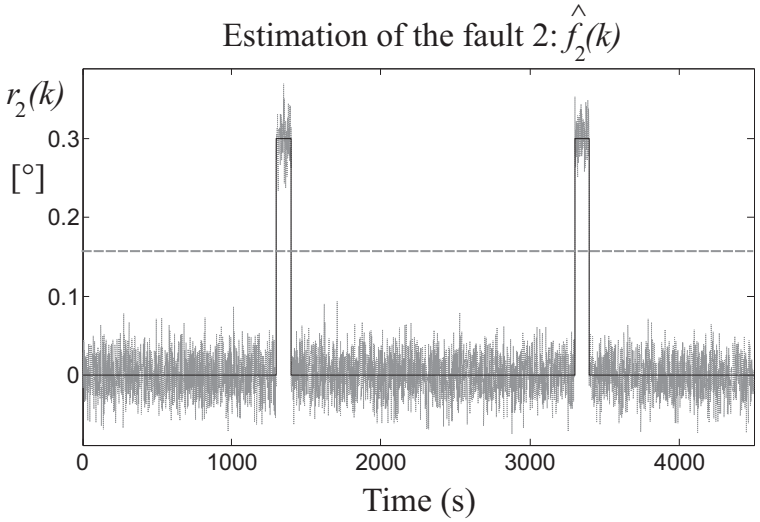

Figure 11. Fault 2 estimator residual $r_2(k)$ (continuous lines) and its threshold level (dotted line).

the FMEA procedure of Section 3.2 and the considered fault cases. The selected architecture of the networks involves two layers, namely the hidden layer and the output layer. The number of neurons in the hidden layer has been fixed to $n_h = 16$. A number of $d_u = d_y = 5$ has been chosen for the input–output delays. Similarly to the fuzzy models, the neural networks modelling capabilities have been tested in terms of RMSE and the results are reported in Table 8 in fault–free conditions for three different data sets (training, validation and test).

Table 8. Neural network performance in terms of RMSE %.

Fault Estimator	1	2	3
RMSE % (training set)	0.0089	0.0091	0.0092
RMSE % (validation set)	0.0091	0.0093	0.0095
RMSE % (validation set)	0.0106	0.0104	0.0123

The fault detection task is achieved by comparing the residual with a fixed optimised threshold, as for the case of the fuzzy estimators, as reported in Table 9.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2017 doi:10.20944/preprints201705.0004.v1

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

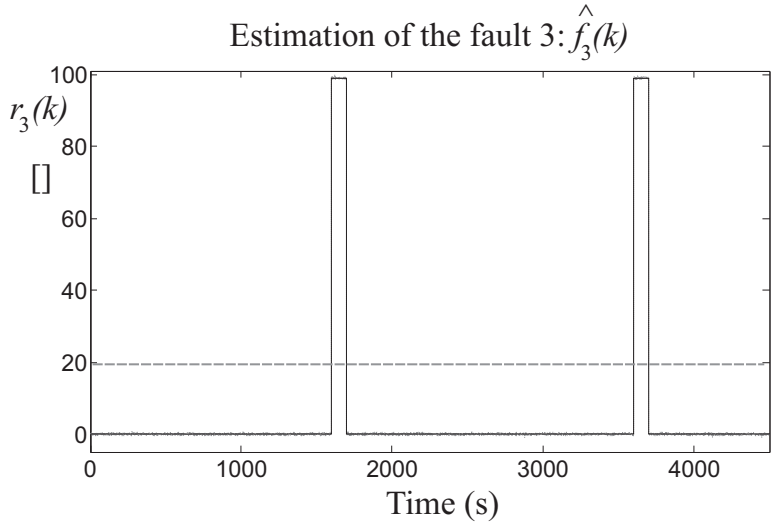

Figure 12. Fault 3 estimator residual $r_3(k)$ (continuous lines) and its threshold level (dotted line).

Table 9. The optimal values of the parameter δ used for fault diagnosis purposes by the designed residual generators.

Residual $r_i(k) = \hat{f}_i(k)$	δ **for the fuzzy estimators**	δ **for the neural network estimators**
1	2.15	2.34
2	2.23	2.45
3	2.34	2.89

5.3. Fault Diagnosis NLGA Simulations

This section describes the design and the simulations of the NLGA adaptive filters applied to the wind park benchmark. In particular, the results achieved from the estimation of the disturbance terms appearing in Eq. (38) are firstly presented. Once the disturbance decoupling has been achieved, the performances of the NLGA fault estimators are reported.

In more detail, the i–th C_p–map entering into the aerodynamic model of each of the 9 wind turbines of the wind farm described in Section 2 has been approximated by using the two–dimensional polynomial in the form of Eq. (44):

$$\hat{C}_p\left(\lambda_i,\, \beta_i\right) = 0.010\,\lambda_i^2 + 0.0003\,\lambda_i^3\,\beta_i - 0.0013\,\lambda_i^3 \tag{44}$$

for the i–th turbine of the wind park. More details they can be found in [32]. By following the same procedure, the second disturbance term representing the $p_d(x)$ function in Eq. (38) is described by the polynomial $\hat{C}_{p_i}$ representing the wind wake from the j–th turbine of the park affecting the i–th turbine:

$$p_d(x) = 0.0027\,\lambda_j^2\,\beta_j - 0.0011\,\lambda_j^2 \tag{45}$$

with $i \neq j$. In general, note that the expression of both the $\hat{C}_{p_i}$–map and the disturbance term $p_d(x)$ in Eq. (45) depend on the i–th wind turbine of the wind farm.

It is also worth observing that the suggested scheme provides the analytical description of the disturbance effects due to all uncertainties, and not only the errors due to C_p entry changes and the wind wake interferences among the wind turbines. However, since these terms are used for the fault estimation filter design, any kind of uncertainty must be modelled. A similar approach was proposed *e.g.* in [15] but developed only for linear state–space models. Under these considerations, the uncertainty distribution description $p_d(x)$ for the nonlinear model of Eq. (38) is identified using the input–output data from the wind turbines of the wind farm. The general assumption holding for this

case is that the model–reality mismatch is varying more slowly that the disturbance signals, such as d. Another important point regards the fact that the $p_d(x)$ estimation aims at describing the structure of the uncertainty, which should not depend on the wind size uncertainty. Only the so–called 'directions' of the disturbance represent the important effect for disturbance decoupling, *i.e.* the $p_d(x)$ term, and not the 'amplitude' of the uncertainty, *i.e.* the size of the disturbance d.

The designed NLGA adaptive filters of Eqs. (41), (42), and (43) provide the estimate the magnitude of the different faults acting on the the wind farm benchmark, as shown in Section 4. With reference to the overall input–affine model of the wind farm described in Section 2, the following terms can be determined:

$$n(x) = \begin{bmatrix} -\frac{\rho A}{2J}\, 0.0010\, R^3\, x_1^2 - \frac{1}{J}\, x_2 \\ -p_{gen}\, x_2 \end{bmatrix} \tag{46}$$

$$g(x) = \begin{bmatrix} 0 & \frac{\rho A}{2J}\, 0.0003\, R^3\, x_2^2 \\ p_{gen} & 0 \end{bmatrix} \tag{47}$$

and:

$$\ell(x) = \begin{bmatrix} 0 & \frac{\rho A}{2J}\, 0.0003\, R^3\, x_1^2 \\ 0 & 0.0001 \end{bmatrix} \tag{48}$$

with reference to the i–th turbine, with the understanding that the subscript i is dropped. Moreover, $p_d(x)$ is defined as:

$$p_d(x) = \begin{bmatrix} \frac{\rho A}{2J}\, 0.0010\, R^2\, x_1 & 0.0011 \\ 0.0002 & \frac{\rho A}{2J}\, 0.0027\, x_2 \end{bmatrix} \tag{49}$$

In the case of the model of Eq. (38), and recalling Eqs. (49), (48), and (47), it results that:

$$S_0 = \bar{P} = \mathrm{cl}\left(p_d(x)\right) \equiv p_d(x) \tag{50}$$

If $\ker\{dh\} = \emptyset$, it follows that $\Sigma_*^P = \bar{P}$ as $\bar{S}_0 \cap \ker\{dh\} = \emptyset$. On the other hand, it is necessary to compute the expression $\left(\Sigma_*^P\right)^\perp = (\bar{P})^\perp$. However, it is worth noting that, for the case under investigation, the determination of the codistribution $\left(\Sigma_*^P\right)^\perp = (\bar{P})^\perp$ is enhanced due to the structure of Eq. (50).

Finally, as an example, the design of the NLGA adaptive filter for the reconstruction of the fault case 2 is based on the expression of Eq. (51):

$$\dot{\tilde{y}}_{1s} = M_1 \cdot f + M_2 \tag{51}$$

where:

$$\begin{cases} M_1 &= 0.8\, x_1^2 - 0.036\, x_1 \\ M_2 &= 1.02\, x_2^2 + 15.7\, x_2 - 0.3\, x_1^3 + 0.77\, x_1^2 \end{cases} \tag{52}$$

The design of the NLGA adaptive filters for the reconstruction of the faults for the cases 1 and 3 is based on a different selection of the vector of Eq. (48), which leads to other expressions for the filter of Eq. (51). As an example, for the fault case 2 described in Section 2.3, the nonlinear filter for the reconstruction of f_2 decoupled from the disturbance d representing the effect of both the wind $v_w(t)$ and the wake $v_{w,m}$ signals has the form of Eq. (41). After a suitable choice of the parameters in Eqs. (41), (42), and (43) the nonlinear filter provided an accurate estimate $\hat{f}_2(k)$ of the fault size, with minimal detection delay.

Finally, also the model capabilities of the 3 NLGA estimators with disturbance decoupling are evaluated in terms of RMSE %, as summarised in Table 10, thus demonstrating the superiority of the model–based approach with respect to the data–driven methods.

The results achieved in Table 10 highlights the efficacy of the NLGA methodology with disturbance decoupling with respect to both the fuzzy and the neural network estimators.

Peer-reviewed version available at Energies 2017, 10, 866; doi:10.3390/en10070866

Table 10. RMSE of the NLGA fault estimator with disturbance decoupling.

Fault Case	Fault 1 Estimator	Fault 2 Estimator	Fault 3 Estimator
RMSE (%)	0.0007	0.0008	0.0009

5.4. 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 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 fault (detection/isolation) and the number of simulated faults;
- **True Fault diagnosis Rate** (TFR): the ratio between the number of correctly detected/isolated faults and the number of simulated faults (complementary to MFR);
- **Mean Fault diagnosis Delay** (MFD): the delay time between the fault occurrence and the fault detection/isolation.

A proper Monte–Carlo analysis has been performed in order to compute these indices and to test the robustness of the considered fault diagnosis schemes. Indeed, the Monte–Carlo 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, extensive simulations based on a set of 10000 Monte–Carlo runs has been executed, during which realistic wind turbine uncertainties have been considered. Some meaningful variables have been modelled as Gaussian white stochastic processes with nominal values and standard deviations corresponding to realistic error values summarised in Table 11.

Table 11. Monte–Carlo analysis parameter variations.

Parameter	**Nominal Value**	**Error Value**
ρ	$1.225 Kg/m^3$	$\pm 20\%$
J	$7.794 \times 10^6 Kg/m^3$	$\pm 30\%$
C_p	C_{p0}	$\pm 50\%$

The comparative analysis results are reported in Table 12. In particular, the different approaches to the fault diagnosis of the wind farm benchmark model, *i.e.* the fuzzy, the neural network and the NLGA adaptive filter fault estimators, are shown.

Table 12. Comparison of the fault diagnosis results with the different fault diagnosis strategies.

Fault Case	**Index**	**Fuzzy Systems**	**Neural Networks**	**NLGA Filters**
1	FAR	0.0010	0.0010	0.0006
	MFR	0.0010	0.0010	0.0007
	TFR	0.9990	0.9990	0.9994
	MFD (s)	0.02	0.01	0.007
2	FAR	0.0010	0.2280	0.0004
	MFR	0.0030	0.0010	0.0005
	TFR	0.9970	0.9990	0.9996
	MFD (s)	0.08	0.08	0.008
3	FAR	0.0030	0.0010	0.0005
	MFR	0.0080	0.0010	0.0006
	TFR	0.9920	0.9990	0.9995
	MFD (s)	0.02	0.01	0.006

The results show the overall efficacy of the proposed fault diagnosis solutions. In general, both the fuzzy and the neural network estimators seem to achieve quite interesting results, and they have

a noteworthy performance level considering the MFD. Also the FAR and MFR are quite low, and in particularly neural networks present very low values of MFR for all the considered faults. However, for both fuzzy and neural networks fault diagnosis design, optimisation stages are required, for example for the selection of the optimal thresholds. On the other hand, the NLGA adaptive estimators definitely show the best results, with respect to the MFD index and the least TFR, with respect to the other fault diagnosis methodologies. However, the design complexity is much higher than the other approaches, and in some cases the analytic solution cannot be achieved.

Finally, the achieved results also lead to further considerations: the proper choice of the design parameters can lead to the least FAR and MFR, with very high TDR, and minimal MFD. Moreover, the Monte–Carlo analysis validates the robustness and the estimation convergence properties of the proposed fault diagnosis schemes, with respect to error, noise and uncertainty.

6. Conclusion

The paper proposed different solutions to the problem of earlier fault detection and diagnosis with application to a wind farm benchmark. The proposed design was based on fault estimators relying on data–driven approaches, as they represented effective tools for coping with a poor analytical knowledge of the system dynamics, together with noise and disturbances. In particular, the data–driven proposed solutions exploited fuzzy systems and neural networks used to describe the strongly nonlinear relationships between the wind farm measurements and its faults. The chosen fuzzy prototype and network architecture belongs to the nonlinear autoregressive with exogenous input 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 farm system. The achieved performances were compared with those of a different fault diagnosis strategy, based on analytical modelling of the physical laws, and relying on a nonlinear disturbance decoupling methodology. Moreover, a Monte–Carlo analysis served to analyse the robustness and reliability features of the proposed solutions against 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 schemes, and possibly with application to real wind turbine systems.

Sample Availability: The software simulation codes for the proposed control strategies and the simulated wind farm benchmark are available from the authors in the MATLAB and Simulink environments.

Acknowledgments: The research works have been supported by the FAR2016 local fund from the University of Ferrara. On the other hand, the costs to publish in open access have been covered by the FIR2016 local fund from the University of Ferrara.

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

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

References

1. Odgaard, P.; Patton, R. FDI/FTC wind turbine benchmark modelling. Workshop on Sustainable Control of Offshore Wind Turbines; Patton, RJ, Ed, 2012.
2. Odgaard, P.F.; Stoustrup, J. Fault tolerant wind farm control. A benchmark model. Control Applications (CCA), 2013 IEEE International Conference on. IEEE, 2013, pp. 412–417.
3. Odgaard, P.F.; Stoustrup, J. A benchmark evaluation of fault tolerant wind turbine control concepts. *Control Systems Technology, IEEE Transactions on* **2015**, *23*, 1221–1228.
4. Chen, W.; Ding, S.X.; Sari, A.; Naik, A.; Khan, A.Q.; Yin, S. Observer-based FDI schemes for wind turbine benchmark. Proceedings of IFAC world congress, 2011, Vol. 18, pp. 7073–7078.
5. Gong, X.; Qiao, W. Bearing fault diagnosis for direct-drive wind turbines via current-demodulated signals. *Industrial Electronics, IEEE Transactions on* **2013**, *60*, 3419–3428.

6. Parker, M.A.; Ng, C.; Ran, L. Fault-tolerant control for a modular generator–converter scheme for direct-drive wind turbines. *Industrial Electronics, IEEE Transactions on* **2011**, *58*, 305–315.
7. Odgaard, P.F.; Stoustrup, J. Results of a wind turbine FDI competition. 8th IFAC symposium on fault detection, supervision and safety of technical processes, 2012, pp. 102–107.
8. Odgaaard, P.F.; Shafiei, S.E. Evaluation of Wind Farm Controller based Fault Detection and Isolation. *IFAC-PapersOnLine* **2015**, *48*, 1084–1089.
9. De Persis, C.; Isidori, A. A geometric approach to nonlinear fault detection and isolation. *Automatic Control, IEEE Transactions on* **2001**, *46*, 853–865.
10. Castaldi, P.; Mimmo, N.; Simani, S. Differential Geometry Based Active Fault Tolerant Control for Aircraft. *Control Engineering Practice* **2014**, *32*, 227–235. Invited Paper. DOI:10.1016/j.conengprac.2013.12.011.
11. Simani, S.; Castaldi, P. Active actuator fault-tolerant control of a wind turbine benchmark model. *International Journal of Robust and Nonlinear Control* **2014**, *24*, 1283–1303.
12. Jensen, N.O. *A note on wind generator interaction*; 1983.
13. Johnson, K.E.; Pao, L.Y.; Balas, M.J.; Fingersh, L.J. Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture. *Control Systems, IEEE* **2006**, *26*, 70–81.
14. Fantuzzi, C.; Simani, S.; Beghelli, S.; Rovatti, R. Identification of piecewise affine models in noisy environment. *International Journal of Control* **2002**, *75*, 1472–1485.
15. Chen, J.; Patton, R.J. *Robust model-based fault diagnosis for dynamic systems*; Vol. 3, Springer Science & Business Media, 2012.
16. Stamatis, D.H. *Failure mode and effect analysis: FMEA from theory to execution*; ASQ Quality Press, 2003.
17. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on* **1985**, pp. 116–132.
18. Babuška, R. *Fuzzy modeling for control*; Vol. 12, Springer Science & Business Media, 2012.
19. Fantuzzi, C.; Rovatti, R. On the approximation capabilities of the homogeneous Takagi-Sugeno model. Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on. IEEE, 1996, Vol. 2, pp. 1067–1072.
20. Rovatti, R. Takagi-Sugeno models as approximators in sobolev norms: the siso case. Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on. IEEE, 1996, Vol. 2, pp. 1060–1066.
21. Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: a review. *ACM computing surveys (CSUR)* **1999**, *31*, 264–323.
22. 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.
23. Graaff, A.J.; Engelbrecht, A.P. Clustering data in stationary environments with a local network neighborhood artificial immune system. *International Journal of Machine Learning and Cybernetics* **2012**, *3*, 1–26.
24. Bezdek, J.C. *Pattern recognition with fuzzy objective function algorithms*; Springer Science & Business Media, 2013.
25. Simani, S.; Fantuzzi, C.; Rovatti, R.; Beghelli, S. Parameter identification for piecewise-affine fuzzy models in noisy environment. *International Journal of Approximate Reasoning* **1999**, *22*, 149–167.
26. Beghelli, S.; Guidorzi, R.P.; Soverini, U. The Frisch scheme in dynamic system identification. *Automatica* **1990**, *26*, 171–176.
27. Haykin, S.S.; Haykin, S.S.; Haykin, S.S.; Haykin, S.S. *Neural networks and learning machines*; Vol. 3, Pearson Education Upper Saddle River, 2009.
28. Liu, G.P. *Nonlinear identification and control: a neural network approach*; Springer Science & Business Media, 2012.
29. Medsker, L.; Jain, L.C. *Recurrent neural networks: design and applications*; CRC press, 1999.
30. Marquardt, D.W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics* **1963**, *11*, 431–441.
31. Hagan, M.T.; Menhaj, M.B. Training feedforward networks with the Marquardt algorithm. *Neural Networks, IEEE Transactions on* **1994**, *5*, 989–993.
32. Simani, S.; Castaldi, P. Estimation of the power coefficient map for a wind turbine system. Proceedings of the 9th European Workshop on Advanced Control and Diagnosis (ACD'11), 2011, number paper 13, pp. 1–7.
33. De Persis, C.; Isidori, A. On the observability codistributions of a nonlinear system. *Systems & control letters* **2000**, *40*, 297–304.

34. Castaldi, P.; Geri, W.; Bonfè, M.; Simani, S.; Benini, M. DESIGN OF RESIDUAL GENERATORS AND ADAPTIVE FILTERS FOR THE FDI OF AIRCRAFT MODEL SENSORS. *Control Engineering Practice* **2009**. ACA'07 – 17th IFAC Symposium on Automatic Control in Aerospace Special Issue. Publisher: Elsevier Science. ISSN: 0967–0661. doi:10.1016/j.conengprac.2008.11.006.