Fault Detection and Isolation for a Wind Turbine Benchmark using a mixed Bayesian/Set-membership Approach☆

Rosa M. Fernandez-Canti[a], Joaquim Blesa[a,b], Sebastian Tornil-Sin[a], Vicenç Puig[a,b,*]

[a] *Research Center for Supervision, Safety and Automatic Control (CS2AC).*
Rambla Sant Nebridi, s/n, 08022 Terrassa (Spain).
[b] *Institut de Robòtica i Informàtica Industrial (CSIC-UPC).*
Carrer LLorens Artigas, 4-6, 08028 Barcelona (Spain).

Abstract

This paper addresses the problem of fault detection and isolation of wind turbines using a mixed Bayesian/Set-membership approach. Modeling errors are assumed to be unknown but bounded, following the set-membership approach. On the other hand, measurement noise is also assumed to be bounded, but following a statistical distribution inside the bounds. To avoid false alarms, the fault detection problem is formulated in a set-membership context. Regarding fault isolation, a new fault isolation scheme that is inspired on the Bayesian fault isolation framework is developed. Faults are isolated by matching the fault detection test results, enhanced by a complementary consistency index that measures the certainty of not being in a fault situation, with the structural information about the faults stored in the theoretical fault signature matrix. The main difference with respect to the classical Bayesian approach is that only models of fault-free behavior are used. Finally, the proposed FDI method is assessed against the wind turbine FDI benchmark proposed in the literature, where a set of realistic fault scenarios in wind turbines are proposed.

Keywords: Fault detection and isolation, Bayesian reasoning, set-membership approaches, wind turbine benchmark, uncertainty

1. Introduction

Wind turbines stand for a growing part of power production. The future of wind energy passes through the installation of offshore wind farms. In such locations, a nonplanned maintenance is very costly. Reducing the cost of wind energy is a key factor in driving successful growth of the wind energy sector. One way of reducing this cost is to use more refined control systems to balance load reduction and power production in an optimal way [1, 2]. Another way of reducing the costs is developing wind turbines that require less scheduled and especially nonscheduled service and have less downtime due to failure [3]. Therefore, a fault-tolerant control (FTC) system that is able to maintain the wind turbine connected after the occurrence of certain faults can avoid major economic losses [4]. An important part of an active FTC system is the implementation of a Fault Detection and Isolation (FDI) system that is able to detect, isolate and, if possible, estimate the faults [5]. Among the fault diagnosis

[*] Corresponding author.
Email address: vicenc.puig@upc.edu (Vicenç Puig)

methodologies, model-based fault diagnosis is the most developed from a conceptual point of view [6, 5, 7, 8].

The application of model-based fault diagnosis in wind turbines has recently been addressed. Revising the literature, the first approaches proposed methods range from Kalman filters [9] to observers [10] and parity equations [11]. The growing interest in wind turbines, coming both from the academia and the industry, has motivated the proposal of a wind turbine benchmark for FDI and FTC, which has increased considerably the research results. This benchmark is based on a realistic generic three blade horizontal variable speed wind turbine with a full converter coupling that contains the most common faults reported in practice [12]. This benchmark has been used in an international competition on FDI and FTC in Wind Turbines aiming at finding the best schemes to diagnose and handle the different faults proposed. According to [12, 13, 14], the most promising solutions proposed are: Data-driven [15], Gaussian Kernel Support Vector Machine [16], Estimation [17], Up-Down Counters [18], Combined Observer and Kalman Filter [19], General Fault Model [20], Fuzzy Models [21], online identification [22] and Set-Membership/Virtual Sensors-Actuators [23, 24].

Fault detection and isolation using statistical methods has been addressed in the literature, starting from the seminal works of Baseville and Nikiforov [25] to the most recent works of Nyberg [26] and [27]. These methods rely on the statistical hypothesis theory for detecting and isolating the fault and they are particularly well suited to take into account measurement noises. As discussed in [28], when detecting faults under a statistical framework, i.e. using Gaussian distributions to model the noise, the selection of the detection threshold is quite critical and could imply false alarms because of the distributions tails. For this reason, [28] suggests to use large thresholds such that false alarms could be avoided. On the other hand, fault detection and isolation has also been considered under a set-membership framework, see for instance [29] or [30]. This framework is more appropriate to deal with modeling errors and its main property is that false alarms are avoided in fault detection. Comparing both approaches, set-membership is appealing for fault detection since false alarms are avoided, but the statistical framework provides more information that can be used for fault detection and isolation purposes. Thus, both approaches are complementary if we consider that one of them is more appropriate to deal with modeling errors whereas the other is more appropriate to deal with measurement noise. However, such a combination has not been proposed in the literature yet, up to the authors' knowledge.

Following the previous discussion, this paper proposes to solve the fault detection and isolation problem using a mixed Bayesian/Set-membership approach. Modeling errors are assumed to be unknown but bounded, following the set-membership approach. On the other hand, measurement noise is described by statistical distributions but with hard bounds, following the idea proposed in [28] as discussed above. Hence, the fault detection problem can be formulated in a deterministic set-membership context. Regarding fault isolation, a new fault isolation scheme that is inspired on the Bayesian fault isolation framework first introduced by [27] is developed. Faults are isolated by matching the fault detection test results, enhanced by a consistency index that measures the certainty of not being in a fault situation, with the structural information about the faults stored in the theoretical fault signature matrix. The main difference with respect to the approach introduced in [27] is that only models of fault-free behavior need to be used.

This paper is organized as follows: Section 2 presents the FDI wind turbine benchmark used as case study in the paper. Section 3 defines the consistency index and describes the residuals evaluation procedure that is to be used for fault detection. Section 4 develops the proposed fault isolation procedure and summarizes the whole fault detection and isolation methodology by means of an algorithm. Section 5 presents the results of the application of the fault detection and isolation approach to the FDI wind turbine benchmark. Finally, Section 6 concludes the paper.

2. Wind turbine benchmark

Wind turbines generate electrical energy from the wind kinetic energy. The wind turbine described in the FDI/FTC benchmark proposed in [12] is a three-blade horizontal-axis variable-speed wind turbine with a full converter coupling. The basic operation principle is that the wind energy is captured by the blades and transformed into mechanical rotational energy through the rotor and the shaft. This energy conversion can be optimized by changing the aerodynamics of the turbine by pitching the blades or by controlling the relative rotational speed of the turbine against the wind speed. The mechanical energy is in turn converted into electrical energy by a generator fully coupled to a converter. Between the rotor and the generator, a drive train is used to increase the rotational speed from the rotor to the generator. The converter can be used to set the generator torque, which consequently can be used to control the rotational speed of the generator as well as the rotor. The objective of the overall control system is to track a power reference.

A block diagram of the wind turbine and its system-level control is presented in Figure 1. The main subsystems are: Blade & Pitch, Drive Train, Generator & Converter and the Controller. The main variable for the operation of the turbine is the wind speed v_ω, that acts as a disturbance from the control perspective. The control reference is the power reference P_r, that in practice is selected according to the wind speed level by an outer loop controller (details can be found in [12]). The controlled inputs are the pitch positions of each blade, β_1, β_2 and β_3, and the converter torque τ_g, being β_r (common reference for the three blades) and $\tau_{g,r}$ the associated references. The pitch position of each blade is measured using two sensors to ensure physical redundancy: $\beta_{1,m1}, \beta_{1,m2}, \beta_{2,m1}, \beta_{2,m2}, \beta_{3,m1}$, $\beta_{3,m2}$. The generator and rotor speeds are also measured with two sensors each: $\omega_{r,m1}$, $\omega_{r,m2}$, $\omega_{g,m1}$, $\omega_{g,m2}$.

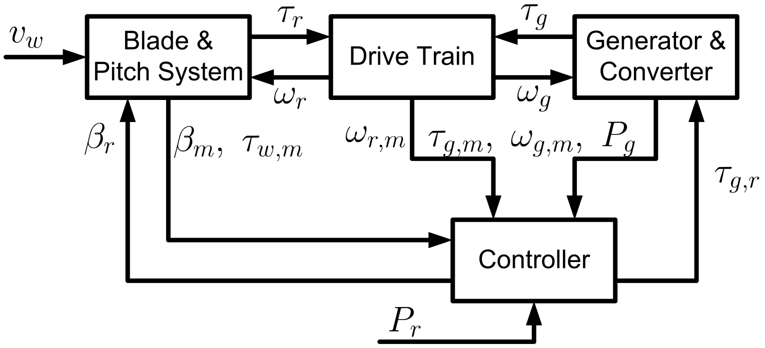

Figure 1: Block diagram of the wind turbine.

Table 1: Wind Turbine Fault Description

Fault	Fault description	Type	Value	Period
f_1	Change in pitch 1 measurement	Fixed Value	$\beta_{1,m1} = 5^o$	2000s-2100s
f_2	Change in pitch 2 measurement	Gain factor	$\beta_{2,m2} = 1.2\beta_{2,m2}$	2300s-2400s
f_3	Change in Pitch 3 measurement	Fixed Value	$\beta_{3,m1} = 10^o$	2600s-2700s
f_4	Change in Rotor speed sensor	Fixed value	$\omega_{r,m1} = 1.4 rad/s$	1500s-1600s
f_5	Change in Rotor and generator speed measurements	Gain factor	$\omega_{r,m2} = 1.1\omega_{r,m2}$ $\omega_{g,m2} = 0.9\omega_{g,m2}$	1000s-1100s
f_6	Parameter abrupt change in pitch 2	Changed dynamics	$\omega_{n2} = 11.11 \rightarrow \omega_{n2} = 5.73$ $\zeta_2 = 0.6 \rightarrow \zeta_2 = 0.45$	2900s-3000s
f_7	Parameter slowly change in pitch 3	Changed dynamics	$\omega_{n3} = 11.11 \rightarrow \omega_{n3} = 3.42$ $\zeta_3 = 0.6 \rightarrow \zeta_3 = 0.9$	3400s-3500s
f_8	Offset in converter system	Offset	$\tau_g = \tau_g + 2000Nm$	3800s-3900s

For each blade, the hydraulic pitch system can be modeled as [31]:

$$\frac{\beta(s)}{\beta_r(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1}$$

where $\beta(s)$ and $\beta_r(s)$ are the pitch angle and its reference, and ω_n and ζ are the natural frequency and the damping ratio of the pitch actuator model.

The generator/converter dynamics can be modeled by:

$$\frac{\tau_g(s)}{\tau_{g,r}(s)} = \frac{\alpha_{gc}}{s + \alpha_{gc}} \tag{2}$$

where τ_g and $\tau_{g,r}$ are the generator torque and its reference and α_{gc} is the generator and converter model parameter.

The power produced by the generator P_g depends on the rotational speed of the generator ω_g and the applied load τ_g:

$$P_g(t) = \eta_g \omega_g(t) \tau_g(t) \tag{3}$$

where η_g is the generator efficiency.

The drive train model consists of a low-speed shaft and a high-speed shaft having inertias J_r and J_g, and friction coefficients B_r and B_g, respectively. The shafts are interconnected by a transmission having a gear ratio N_g and an efficiency η_{dt}, combined with a torsion stiffness K_{dt}, and a torsion damping B_{dt}. The model is described by the following three differential equations [32]:

$$\dot{\omega}_r(t) = -\frac{(B_{dt} + B_r)}{J_r}\omega_r(t) + \frac{B_{dt}}{N_g J_r}\omega_g(t) - \frac{K_{dt}}{J_r}\theta_\Delta(t) + \frac{\tau_r(t)}{J_r} \tag{4}$$

$$\dot{\omega}_g(t) = \frac{\eta_{dt} B_{dt}}{N_g J_g}\omega_r(t) - \left(\frac{\eta_{dt} B_{dt}}{N_g^2 J_g} + \frac{B_g}{J_g}\right)\omega_g(t) + \frac{\eta_{dt} K_{dt}}{N_g J_g}\theta_\Delta(t) - \frac{\tau_g(t)}{J_g} \tag{5}$$

$$\dot{\theta}_{\Delta}(t) = \omega_r(t) - \frac{\omega_g(t)}{N_g} \tag{6}$$

where ω_r is the rotor speed, ω_g is the generator speed, θ_Δ is the torsion angle of the drive train, τ_r is the aerodynamic torque and τ_g is the generator torque.

The values for all the model parameters and details about the controller can be found in [12].

In this paper, the different faults proposed in the FDI/FTC benchmark [12] will be considered, as resumed in Table 1.

3. Residual Generation and Evaluation for Fault Detection

3.1. Residual generation

Considering the wind turbine model and the set of available sensors defined in the wind turbine FDI benchmark [12], the application of the structural analysis leads to a set of analytical redundancy relations (ARRs). For each ARR, a residual is obtained as

$$r(k, \theta) = y(k) - y_m(k) \tag{7}$$

where

- $y(k)$ is the output sensor measurement.
- $y_m(k)$ is the the estimation provided by the ARR expression and it is affected by two sources of uncertainty: $y_m(k) = \hat{y}(k, \theta) + e(k)$, where:
  - $\hat{y}(k, \theta)$ depends of an unknown parameter vector $\theta \in \Theta$ of dimension $n_\theta \times 1$ and Θ is the parameter uncertainty set.
  - $e(k)$ is the additive error term which is unknown but it is assumed to be zero mean and bounded by a constant, $|e(k)| \leq \delta$. Within the bounds $[-\delta, \delta]$, a probability distribution $p_e(e)$ is assumed as well.
- k is the discrete-time sample.

According to [7], the residual generator can be derived from the input/output form of the ARR as

$$\hat{y}(k, \theta) = G(q^{-1}, \theta)u(k) + H(q^{-1}, \theta)y(k) \tag{8}$$

where $u(k)$ is the input and the particular expressions of $G(q^{-1}, \theta)$ and $H(q^{-1}, \theta)$ depend on the approach used to obtain $\hat{y}(k, \theta)$, e.g. a parity equation or an observer (see [29] for more details).

3.2. Bayesian set-membership framework

The residual (7) can be used to detect faults. Thus, a fault is reported if the deviation of the measurement $y(k)$ cannot be explained by a no faulty model which takes into account both the additive noise and the parameters uncertainty. If a fault occurs, the residual (7) is violated, i.e. its magnitude (obtained considering the model uncertainty) exceeds the pre-specified noise threshold δ.

The effect of the uncertain parameters θ and the additive noise $e(k)$ on the residual (7) can be bounded using the Bayesian set-membership framework introduced in [33]. This framework relies on the following assumptions:

- The parameter prior distribution $p(\theta)$ is assumed to be flat over the considered parameter set Θ, that is, all models θ in the uncertainty set Θ enjoy, *a priori*, the same probability level.

- For a fixed θ, the value of $\hat{y}(k,\theta)$, $\forall k$, is deterministic and it can be computed. Since the observation $y(k)$ is also known, the only stochastic term in (7) is the additive noise $e(k)$. Therefore, we assume that the residual probability distribution $p_r(r|\theta)$ coincides in form with the error probability distribution $p_r(r|\theta) \equiv p_e(e)$.

- Following the equation-error assumption introduced in [34], we assume that the additive noise samples $e(k)$ are i.i.d. (independent and identically distributed). This allows us to estimate the joint probability distribution $p_r(r|\theta)$ numerically and sample-to-sample,

$$p_r(r|\theta) = \prod_{k=1}^{M} p_r(r(k)|\theta) \tag{9}$$

3.3. Residual evaluation

Regarding the model uncertainty, and without loss of generality, let us consider that the parameter uncertainty set is an axis-aligned box $\mathbf{\Theta}$:

$$\mathbf{\Theta} \triangleq [\theta_1] \times \cdots \times [\theta_{n_\theta}] \tag{10}$$

where $[\theta_j] = [\underline{\theta_j}, \overline{\theta_j}]$, $j = 1, \ldots, n_\theta$, are the interval bounds for each component θ_j of the parameter vector θ, and the operator × represents the Cartesian product. The interval evaluation of this set of models leads to an interval model response $[\hat{y}(k)]$ containing all the possible model responses

$$[\hat{y}(k)] \equiv \left[\underline{\hat{y}}(k), \overline{\hat{y}}(k)\right] \tag{11}$$

where both the estimation $\hat{y}(k,\theta)$ and the measurement $y(k)$ are assumed to be, $\hat{y}(k,\theta), y(k) \in [\hat{y}(k)]$, in the non-faulty case. This interval quantifies the effect of the parameter uncertainty in the model response and it can be computed as

$$\underline{\hat{y}}(k) = \min_{\theta\in\Theta} \hat{y}(k,\theta) \quad \text{and} \quad \overline{\hat{y}}(k) = \max_{\theta\in\Theta} \hat{y}(k,\theta) \tag{12}$$

subject to the equations given by (8). The optimization problems (12) can be solved efficiently by using zonotopic approach described in [35] that since only involves matrix algebraic operations presents a polynomial complexity.

The parameter uncertainty intervals (10) are determined using the optimization procedure described in [36]. This procedure provides the more adjusted prediction intervals (12) that avoid false alarms while minimizing the size of the detectable faults.

The interval model response $[\hat{y}(k)]$ allows us to define the interval residual as $[r(k)] \equiv y(k) - [\hat{y}(k)]$, $[r(k)] = [\underline{r}(k), \overline{r}(k)]$.

Now, to introduce the effect of the additive noise, we compare the interval residual $[r(k)]$ to the interval defined by the additive noise term bounds, $[-\delta, \delta]$. If $[r(k)]$ is totally outside the interval $[-\delta, \delta]$, we decide that there is a fault and we take the fault indicator ϕ equal to one. Otherwise, we assume that at least one model in the uncertainty set is able to explain the observed behavior (in other words, at least one of the models in the uncertainty region is consistent to the measurements) and consequently we decide that there is no fault taking the fault indicator ϕ equal to zero. To illustrate this point, see Figure 2, where the cases (a), (b) and (c) correspond to the no faulty situation whereas the case (d) corresponds to the faulty case.

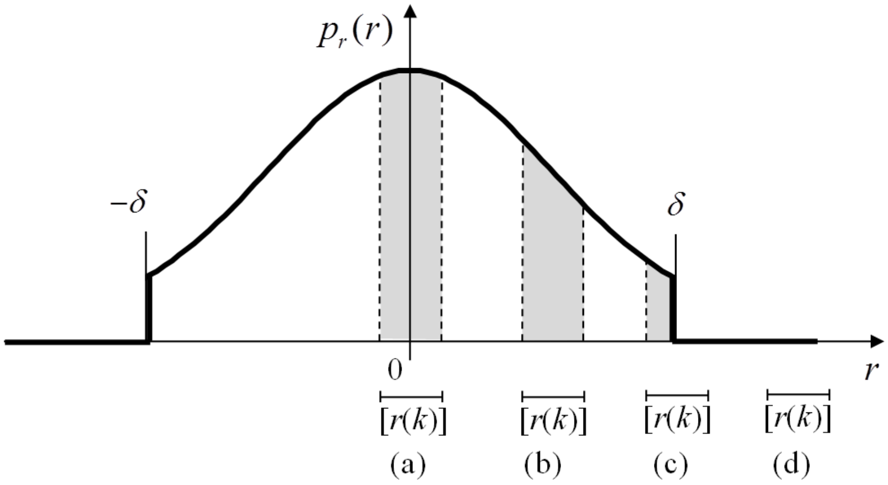

Figure 2: Residual evaluation

The worst-case set-membership viewpoint adopted in the fault detection procedure described above implicitly assumes the no existence of false alarms. Therefore, when we observe $\phi = 1$ we assume that it corresponds to a true fault. The price to pay is that when we observe $\phi = 0$ we cannot be sure that no fault has occurred, i.e., the uncertainty may mask the faults, leading to a certain rate of missing faults. This is a well-known problem in the set-membership approaches [29] and it can induce errors in the fault isolation stage.

In order to solve this problem, in the residual evaluation stage, we propose to produce not only the fault indicator ϕ but also an index α indicating how likely is that the fault indicator $\phi = 0$ (apparently indicating no fault) may correspond to a fault. Note that when we have observed a fault, the α-index is 1.

The α-index can be computed as $\alpha = 1 - \gamma$ where γ is the so-called *consistency index*. The consistency index γ, $0 < \gamma \leq 1$, is a number assigned to each non faulty indicator ($\phi = 0$) that indicates our confidence about if the fault indicator corresponds to the no faulty case or, on the contrary, it may correspond to a missing fault. Note that in the case that we have observed a fault ($\phi = 1$), the consistency index would be $\gamma = 0$. In other words, when we have a fault, the observed behaviour is totally *not consistent* with the uncertainty model and noise level considered.

To illustrate the computation of the consistency index γ, it is assumed that the residual $r(k)$ is zero mean Gaussian distributed inside the hard bounds given by $-\delta$ and δ (see Figure 2). Thus, the set-membership feasibility region (or uncertainty region) lays between the hard bounds given by $-\delta$ and δ. The novelty in this work, compared to the pure set-membership approaches, is that, in order to compute the consistency index γ, we combine the hard bounds with the probability distribution of the residual.

Intuitively, an interval residual located in the center of the uncertainty region (case (a)) is less likely to correspond to a missing fault than an interval which is closer to the region border (case (b)) or even in the border itself (case (c)).

Each of the shadowed areas in Figure 2 can be computed as $A = \int_{\max(-\delta,\underline{r}(k))}^{\min(\overline{r}(k),\delta)} p_r(r)dr$. For the case (a), note that the shadowed area is $A_0 = \int_{-w_r/2}^{w_r/2} p_r(r)dr$, being w_r the width of the interval residual $[r(k)]$. And note that $A_0 \geq A$ for every interval $[r(k)]$ with w_r width.

In these conditions, the consistency index γ associated to the interval residual with width w_r can be computed as

$$\gamma = \frac{A}{A_0} = \frac{\int_{\max(-\delta,\underline{r}(k))}^{\min(\overline{r}(k),\delta)} p_r(r)dr}{\int_{-w_r/2}^{w_r/2} p_r(r)dr} \tag{13}$$

In the case of the Gaussian probability distribution shown in Figure 2, the consistency index is $\gamma = 1$ when the interval residual is totally centered around the mean value of p_r or $[-\delta, \delta] \subset [r(k)]$ and it is $\gamma = 0$ when it is outside the set-membership region. The values $0 < \gamma < 1$ indicate different degrees of consistency. The smaller the consistency index γ is, the larger the probability of a missing fault is. In the case of other distributions, for example nonsymmetric distributions, the area A_0 must be the largest that we can obtain for a fixed w_r, since this corresponds to the region around the nominal model (where the residual probability distribution should present its maximum value).

The whole procedure for the residual evaluation is summarized in Algorithm 1.

Algorithm 1 Residual evaluation.

```
Algorithm ResidualEvaluation($u$,$y$,$[\Theta]$, $\delta$, $p_r$)
      $\gamma \leftarrow 0$
      Obtain $[\hat{y}]$ by means of (8) and (12) using $[\Theta], u, y$
      $[r] \leftarrow y - [\hat{y}]$
      if $[r] \cap [-\delta, +\delta] \neq \emptyset$ then
        $\phi \leftarrow 0$
        Compute $\gamma$ using (13) by considering $p_r$
        $\alpha \leftarrow 1 - \gamma$
      else
        $\phi \leftarrow 1$
        $\alpha \leftarrow 1$
      endif
  return($\phi, \alpha$)
endAlgorithm
```

4. Fault detection and isolation

Fault isolation aims at identifying the faults affecting the system. In standard fault isolation [7], this is carried out on the basis of the observed fault signatures, generated by the detection module, and their relation with all the considered faults. Here, the proposed approach follows the same spirit but the reasoning is not based on logic rules. Instead, a procedure inspired on a Bayesian reasoning framework [27] is used.

As discussed in the introduction, the main difference with the approach proposed in [27] is that in our approach we only use models of nominal (no faulty) behavior whereas in [27] the reasoning is applied not only to the nominal model but also to models corresponding to the different faulty situations. The approach proposed here is possible because of the use of the mixed Bayesian/set-membership framework, introduced in Section 3, which assumes worst-case hard bounds for the noise, thus allowing us to discard false alarms.

The fault isolation can be performed by considering both the fault indicators $\boldsymbol{\phi}^T = (\phi_1, \cdots, \phi_n)$, being n the number of considered residuals, and consistency indexes $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_n)$ and comparing them to a *Fault Signature Matrix* (FSM), denoted as $\boldsymbol{M}$. This matrix exploits the knowledge about the binary relation between the set of fault hypotheses and the set of residuals. An element m_{ij} of $\boldsymbol{M}$ is equal to 1 if the fault f_j affects the computation of the residual r_i; otherwise $m_{ij} = 0$. A column $\boldsymbol{m}_j$ of $\boldsymbol{M}$ is known as a *theoretical fault signature* and indicates which residuals are affected by a given fault f_j. A set of faults is *isolable* if all the columns in $\boldsymbol{M}$ are different (two identical columns indicate two indistinguishable faults).

Since we are looking for faults, we are interested in knowing how close has been the fault indicator $\phi_i = 0$ to be a 1. For this reason, for each residual r_i, instead of taking the consistency index γ_i we take its complementary, $\alpha_i = 1 - \gamma_i$. This way, a value for the *inconsistency index* α_i near to 1 indicates that the fault indicator $\phi_i = 0$ could in fact be an undetected 1 (i.e., a missing fault). And if α_i is small, the probability that $\phi_i = 0$ would correspond to a missing fault is small, too.

4.1. *Computation of the posterior probabilities*

Given the FSM and given the fault indicators $\boldsymbol{\phi}^T = (\phi_1, \cdots, \phi_n)$ with their associated α-indexes, $\boldsymbol{\alpha}^T = (\alpha_1, \cdots, \alpha_n)$, the isolation of a particular fault from the FSM can be performed by means of the computation of each fault posterior probability by applying the Bayes rule:

$$p(f_j|\boldsymbol{\phi}) = \frac{p(\boldsymbol{\phi}|f_j)p(f_j)}{\sum_{l=1}^{m} p(\boldsymbol{\phi}|f_l)p(f_l)} \tag{14}$$

where $f_j, j = 1, ..., m$, are the faults, $p(f_j)$ is the prior probability assigned to fault f_j, $p(f_j|\boldsymbol{\phi})$ is the posterior probability assigned to fault f_j, and $p(\boldsymbol{\phi}|f_j)$ is a value indicating the likelihood that the fault f_j is behind the observed vector $\boldsymbol{\phi}$. This likelihood can be computed for each sample time k as

$$p(\boldsymbol{\phi}|f_j) = \boldsymbol{\alpha}(k)^T \frac{\boldsymbol{m}_j}{n_j} zvf_j(k) \tag{15}$$

being $\boldsymbol{m}_j$ the column in the FSM associated to fault f_j, n_j the number of nonzero values in the m_j and zvf_j is the zero-violation-factor that checks the consistency of the fault to the violated residuals. This factor can be computed as

$$zvf_j = \begin{cases} 0, & \text{if } \exists i \in \{1, ..., n\} \text{ with } m_{ij} = 0 \quad \text{and} \quad \phi_i = 1 \\ 1, & \text{otherwise} \end{cases} \tag{16}$$

The term (16) assumes that we do not have false alarms, therefore given the fault indicator vector $\boldsymbol{\phi}^T = \left(\phi_1, \cdots, \phi_{n_y}\right)$ some faults can be directly discarded. This is the case when the fault indicator for a certain residual is 1 whereas the value for that residual in the corresponding fault column is 0.

Note that the prior and posterior probabilities must satisfy $\sum_{j=1}^{m} p(f_j) = 1$ and $\sum_{j=1}^{m} p(f_j|\boldsymbol{\phi}) = 1$, respectively.

Finally, note that the procedure is intrinsically recursive since the posterior probabilities (14) can be used as prior probabilities in successive iterations, and as long as new measurements become available.

4.2. Example

Let us illustrate the methodology by means of an example. Consider the following FSM,

Table 2: FSM example

	f_1	f_2	f_3	f_4
r_1	1	1	0	0
r_2	0	1	1	1
r_3	1	0	1	0

and consider that, for the three considered residuals, r_i, $i = 1, 2, 3$, we have the fault indicator vector $\boldsymbol{\phi}^T = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$ with the associated indexes $\boldsymbol{\alpha}^T = \begin{pmatrix} 025 & 1 & 0.35 \end{pmatrix}$. Note that the α-index associated to the fault indicator ($\phi_2 = 1$) is $\alpha_2 = 1$. From, the fault indicator vector $\boldsymbol{\phi}$ we already know that there is a fault but we have to decide which one is.

Let us assume that, *a priori*, all four faults present the same probability, i.e., $p(f_1) = p(f_2) = p(f_3) = p(f_4) = 1/4$. Note that $\sum_{j=1}^{4} p(f_j) = 1$.

Now, given the α-indexes and the FSM provided in Table 2, we can compute the likelihood values by applying (15). Note that fault f_1 results directly discarded from the analysis because of the factor zvf_1

$$p(\boldsymbol{\phi}|f_1) = \boldsymbol{\alpha}^T \frac{\boldsymbol{m}_1}{n_1} zvf_1 = \begin{pmatrix} 025 & 1 & 0.35 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \frac{1}{2} \cdot 0 = 0 \tag{17}$$

$$p(\boldsymbol{\phi}|f_2) = \boldsymbol{\alpha}^T \frac{\boldsymbol{m}_2}{n_2} zvf_2 = \begin{pmatrix} 025 & 1 & 0.35 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \frac{1}{2} \cdot 1 = 0.625 \tag{18}$$

$$p(\boldsymbol{\phi}|f_3) = \boldsymbol{\alpha}^T \frac{\boldsymbol{m}_3}{n_3} zvf_3 = \begin{pmatrix} 025 & 1 & 0.35 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \frac{1}{2} \cdot 1 = 0.675 \tag{19}$$

$$p(\boldsymbol{\phi}|f_4) = \boldsymbol{\alpha}^T \frac{\boldsymbol{m}_4}{n_4} zvf_4 = \begin{pmatrix} 025 & 1 & 0.35 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \frac{1}{1} \cdot 1 = 1 \tag{20}$$

The posterior probability values given by (14) are:

$$p(f_2|\boldsymbol{\phi}) = \frac{0.625 \cdot 1/4}{0.625 \cdot 1/4 + 0.675 \cdot 1/4 + 1 \cdot 1/4} = 0.2717 \tag{21}$$

$$p(f_3|\boldsymbol{\phi}) = \frac{0.675 \cdot 1/4}{0.625 \cdot 1/4 + 0.675 \cdot 1/4 + 1 \cdot 1/4} = 0.2934 \tag{22}$$

$$p(f_4|\boldsymbol{\phi}) = \frac{1 \cdot 1/4}{0.625 \cdot 1/4 + 0.675 \cdot 1/4 + 1 \cdot 1/4} = 0.4348 \tag{23}$$

Since the fault f_4 is the most probable fault, we can decide that the fault f_4 is the isolated one. And comparing faults f_2 and f_3, we see that the probability to be the fault behind the observed behavior is slighty larger for fault f_3 (0.29) than for fault f_2 (0.27). This result was expected since the observed 0 in the third residual is closer to be a missing fault ($\phi_3 = 0$, $\alpha_3 = 0.35$) than the observed 0 in the first residual ($\phi_1 = 0$, $\alpha_1 = 0.25$).

One could argue that the posterior probability of f_4 is not so far from the posterior probabilities of f_2 and f_3. It is true, but this result can be explained because we have used only one time sample in the reasoning process. To improve the results, we can use the fact that the method allows updating the posterior probabilities with new measurement data. As long as new measurements are used, one fault tends to one while the others tend to zero. If we want to stop the search before reaching the probability one for an unique fault, we can select a desired precision ϵ for the fault isolation procedure.

The whole fault detection and isolation methodology is described in the next subsection.

4.3. Fault detection and isolation algorithm

The fault detection and isolation methodology presented above can be implemented by means of the Algortihm 2. First, the variables $faultDetected$, $faultIsolated$ and $mostProbableFault$ are initialized. A fault is isolated if its posterior probability is greater or equal to $1 - \epsilon$ being ϵ the desired precision. While a fault has not been isolated, the residual evaluation Algorithm 1 is applied to every system output y_i, $i = 1, ..., n$, providing the fault indicators ϕ_i and α_i indexes. If any fault indicator ϕ_i is activated for the first time, the prior probabilities for all m faults are initialized to $1/m$. After a fault has been detected, the posterior probability of each fault is iteratively computed and the most probable fault at each sample time k is provided. The fault detection and isolation algorithm ends when the posterior probability of one fault is 1 or near 1(greater or equal to $1 - \epsilon$) and, consequently, the posterior probabilities of the other faults are 0 or close to 0 (less or equal to ϵ).

Algorithm 2 Fault detection and isolation

```
faultDetected ← FALSE
faultIsolated ← FALSE
mostProbableFault ← EMPTY
$k \leftarrow 0$
while faultIsolated ≠ TRUE
Obtain input-output data $\{u(k), y(k)\}$ at time instant $k$
for $i = 1$ to $n$
    $(\phi_i(k), \alpha_i(k)) \leftarrow$ ResidualEvaluation($u(k)$,$y_i(k)$,$[\Theta]$, $\delta_i$, $p_{r_i}$)
endfor
if $\exists \phi_i(k) \neq 0$ and faultDetected = FALSE then
    faultDetected ← TRUE
    for $j = 1$ to $m$
        $p(f_j) \leftarrow \frac{1}{m}$
    endfor
endif
if faultDetected=TRUE
    for $j = 1$ to $m$
        Compute $zvf_j(k)$ using (16)
        $p(\boldsymbol{\phi}|f_j) \leftarrow \boldsymbol{\alpha}(k)^T \frac{\boldsymbol{m}_j}{n_j} zvf_j(k)$
    endfor
    $\Sigma \leftarrow 0$
    for $j = 1$ to $m$
        $\Lambda_j \leftarrow p(\boldsymbol{\phi}|f_j)p(f_j)$
        $\Sigma \leftarrow \Sigma + \Lambda_j$
    endfor
    for $j = 1$ to $m$
        $p(f_j|\boldsymbol{\phi}) \leftarrow \frac{\Lambda_j}{\Sigma}$
        $p(f_j) \leftarrow p(f_j|\boldsymbol{\phi})$
    endfor
    mostProbableFault ← max($p(f_1|\boldsymbol{\phi}), ..., p(f_{n_f}|\boldsymbol{\phi})$)
    if $\exists p(f_j|\boldsymbol{\phi}) \geq 1 - \epsilon$ then
        faultIsolated ← TRUE
    endif
endif
$k \leftarrow k + 1$
endwhile
return($f_j$)
```

5. Application and results

5.1. Residual Generation

According to [23], after applying structural analysis [6] with the aid of the SaTool [37] to the set of equations provided in [12], the following set of twelve residuals expressed as reduced observers in input-output form can be obtained:

$$
\begin{aligned}
r_1(k) &= \omega_{r,m1}(k) - \omega_{r,m2}(k) \\
r_2(k) &= \left(1 - \frac{l_{21}q^{-1}}{1-(a_{2,1}-l_{2,1})q^{-1}}\right)\omega_{r,m2}(k) - \frac{b_{2,1}q^{-1}\tau_r(k) + c_{2,1}q^{-1}\tau_{g,m}(k)}{1-(a_{2,1}-l_{2,1})q^{-1}} \\
r_3(k) &= \omega_{g,m1}(k) - \omega_{g,m2}(k) \\
r_4(k) &= \left(1 - \frac{l_{4,1}q^{-1}}{1-(a_{4,1}-l_{4,1})q^{-1}}\right)\omega_{g,m2}(k) - \frac{b_{4,1}q^{-1}\tau_r(k) + c_{4,1}q^{-1}\tau_{g,m}(k)}{1-(a_{4,1}-l_{4,1})q^{-1}} \\
r_5(k) &= \beta_{1,m1}(k) - \beta_{1,m2}(k) \\
r_6(k) &= \left(1 - \frac{l_{6,1}q^{-1} + l_{6,2}q^{-2}}{1-(a_{6,1}-l_{6,1})q^{-1}-(a_{6,2}-l_{6,2})q^{-2}}\right)\beta_{1,m2}(k) \\
&- \frac{(b_{6,1}q^{-1} + b_{6,2}q^{-2})\beta_r(k)}{1-(a_{6,1}-l_{6,1})q^{-1}-(a_{6,2}-l_{6,2})q^{-2}} \\
r_7(k) &= \beta_{2,m1}(k) - \beta_{2,m2}(k) \\
r_8(k) &= \left(1 - \frac{l_{8,1}q^{-1} + l_{8,2}q^{-2}}{1-(a_{8,1}-l_{8,1})q^{-1}-(a_{8,2}-l_{8,2})q^{-2}}\right)\beta_{2,m2}(k) \\
&- \frac{(b_{8,1}q^{-1} + b_{8,2}q^{-2})\beta_r(k)}{1-(a_{8,1}-l_{8,1})q^{-1}-(a_{8,2}-l_{8,2})q^{-2}} \\
r_9(k) &= \beta_{3,m1}(k) - \beta_{3,m2}(k) \\
r_{10}(k) &= \left(1 - \frac{l_{10,1}q^{-1} + l_{10,2}q^{-2}}{1-(a_{10,1}-l_{10,1})q^{-1}-(a_{10,2}-l_{10,2})q^{-2}}\right)\beta_{3,m2}(k) \\
&- \frac{(b_{10,1}q^{-1} + b_{10,2}q^{-2})\beta_r(k)}{1-(a_{10,1}-l_{10,1})q^{-1}-(a_{10,2}-l_{10,2})q^{-2}} \\
r_{11}(k) &= \left(1 - \frac{l_{11,1}q^{-1}}{1-(a_{11,1}-l_{11,1})q^{-1}}\right)\tau_{g,m}(k) - \frac{b_{11,1}q^{-1}\tau_{g,r}(k)}{1-(a_{11,1}-l_{11,1})q^{-1}} \\
r_{12}(k) &= P_{g,m}(k) - \eta_g\omega_{g,m2}(k)\tau_{g,m}(k)
\end{aligned}
$$

where $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$ are model parameters that have to be estimated (η_g is a known coefficient) such that:

$$
\begin{aligned}
\theta = \Big(& a_{2,1} \quad b_{2,1} \quad c_{2,1} \quad a_{4,1} \quad b_{4,1} \quad c_{4,1} \quad a_{6,1} \quad a_{6,2} \quad b_{6,1} \quad b_{6,2} \\
& a_{8,1} \quad a_{8,2} \quad b_{8,1} \quad b_{8,2} \quad a_{10,1} \quad a_{10,2} \quad b_{10,1} \quad b_{10,2} \quad a_{11,1} \quad b_{11,1} \Big)^T
\end{aligned}
\tag{24}
$$

and $l_{i,j}$ are the observer gains. This set of residuals will be used for fault detection and isolation. It must be noticed that a nonlinearity is hidden in some of the residuals due to the use of the variable τ_r, which is estimated using the following relation given in [38]:

$$\tau_r(k) = \rho\pi R^3 C_q(\lambda(k),\beta(k))v_w(k)^2/2. \tag{25}$$

Moreover, SaTool provides the FSM represented in Table 3, which captures the relation between residuals and faults, where a cross 'x' indicates that a given residual is affected by a given fault (according to the notation used in Section 4, $m_{ij} = 1$ where there is a cross, $m_{ij} = 0$ elsewhere).

Notice that this paper is not focused on how to obtain the residuals. Structural methods have actually been used to obtain the residuals for this application, but other methods could be considered and the proposed method could be applied to the resulting set of residuals, e.g. those obtained by [39].

Table 3: Fault signature matrix

r	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
r_1				x	x			
r_2	x	x	x		x	x	x	x
r_3					x			
r_4	x	x	x		x	x	x	x
r_5	x							
r_6	x							
r_7		x						
r_8		x				x		
r_9			x					
r_{10}			x				x	
r_{11}								x
r_{12}					x			

Residuals r_1, r_3, r_5, r_7 and r_9 are static equations that involve the comparison of the value of two different sensors measuring the same variable. In the same way, the residual r_{12} compares the value of a sensor with the one calculated using a static relation with two other sensors. Residuals r_2 and r_4 are obtained from the drive train model. Residuals r_6, r_8 and r_{10} follow from the hydraulic pitch system model. Finally, the residual r_{11} is derived from the converter dynamics model [36].

Previous residuals are affected by noise and parametric uncertainties because residual parameters (24) are estimated using measurements and in some cases with reduced order models. In order to reduce the effect of the noise in the fault detection and isolation procedures, the variables involved in the discretized regressor equations have been filtered by second order low-pass filters. Nominal parameters and uncertain parameter bounds of dynamical residuals have been obtained as described in [36].

Finally, given a set of nonfaulty data, fault-free residual probability distributions p_{r_i}, $i = 1, \ldots, 12$ are computed. Figure 3 depicts the obtained residual probability distributions.

5.2. Fault Detection and Isolation

Once residual parameters and probability distribution functions have been calibrated, the proposed fault detection and isolation algorithm of Section 4 has been applied to the data provided by a realistic wind turbine simulator in the fault scenarios proposed in [12]. For the static residuals: r_1, r_3, r_5, r_7, r_9 and r_{12} the consistency indexes $\gamma_i(k)$ have been computed using

$$\gamma_i(k) = \frac{p_{r_i}(r_i(k))}{max(p_{r_i})} \tag{26}$$

Next, some results are reported. First, the particular results for three faults (4, 6, and 7) of the sequence specified in Table 1 are presented and analyzed in detail to illustrate how the method works. After that, the summarizing results obtained from a more extensive study, a whole set of Monte Carlo simulations taking into account the eight faults under different scenarios, are provided and compared to the ones obtained by other FDI methods that have been applied to the wind turbine FDI/FTC benchmark.

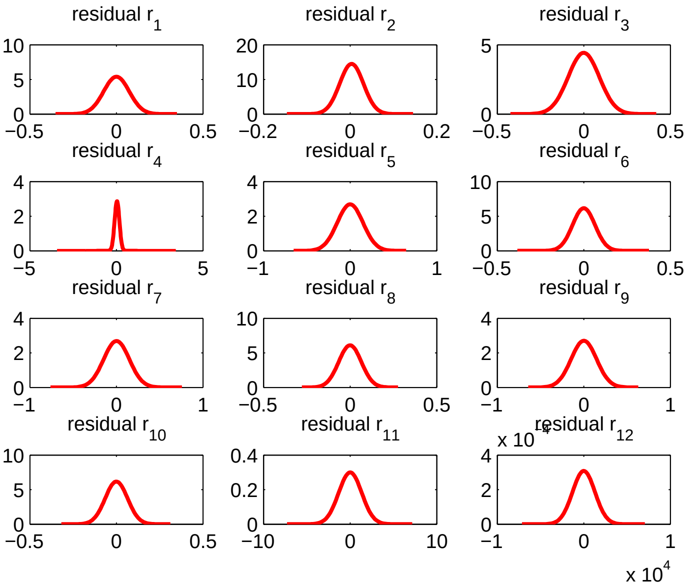

Figure 3: Fault free residual distributions p_{r_i}, $i = 1, ..., 12$

Results for fault f_4

In this scenario, the fault is introduced at instant $t = 1500s$. Only the residual fault indicator ϕ_1 is permanently activated from instant $t = 1500.03s$ during the time that the fault is active and thus $\alpha_1 = 1$ in this time period. Figure 4 shows the evolution of all coefficients α_i, $i = 1, \ldots, 12$, provided by Algorithm 1. Figure 5 depicts the evolution of the fault posterior probabilities $p(f_i|\boldsymbol{\phi})$, $i = 1, \ldots, 12$, provided by Algorithm 2. As it can be seen in Figure 5 only posterior probabilities $p(f_4|\boldsymbol{\phi})$ and $p(f_5|\boldsymbol{\phi})$ are different from zero. The detail in Figure 6, shows how, from the fault detection time $t = 1500.03s$, $p(f_4|\boldsymbol{\phi}) > p(f_5|\boldsymbol{\phi})$ and therefore the most probable fault provided by Algorithm 2 is the fault f_4 (the correct one).

Results for fault f_6

In scenario, the fault occurs at instant $t = 2900s$. Only the residual fault indicator ϕ_8 is permanently activated from instant $t = 2900.06s$ during the time that the fault is active and thus $\alpha_8 = 1$ in this time period. Figure 7 shows the evolution of all coefficients α_i, $i = 1, \ldots, 12$, provided by Algorithm 1 and Figure 8 depicts the evolution of the fault posterior probabilities $p(f_i|\boldsymbol{\phi})$, $i = 1, \ldots, 12$, provided by Algorithm 2. As it can be seen in Figure 8 only posterior probabilities $p(f_2|\boldsymbol{\phi})$ and $p(f_6|\boldsymbol{\phi})$ are different from zero and as it can be observed in Figure 9, from the fault detection time $t = 2900.06s$, $p(f_6|\boldsymbol{\phi}) > p(f_2|\boldsymbol{\phi})$. Therefore, the most probable fault provided by Algorithm 2 is the fault f_6 (the correct one).

Results for fault f_7

In this scenario, the fault appears at instant $t = 3400s$. Only the residual fault indicator ϕ_{10} is permanently activated from time instant $t = 3411.60s$ during the time that the fault is active and thus $\alpha_{10} = 1$ in this time period. Figure 10 shows the evolution of all coefficients α_i, $i = 1, \ldots, 12$, provided by Algorithm 1 and Figure 11 depicts

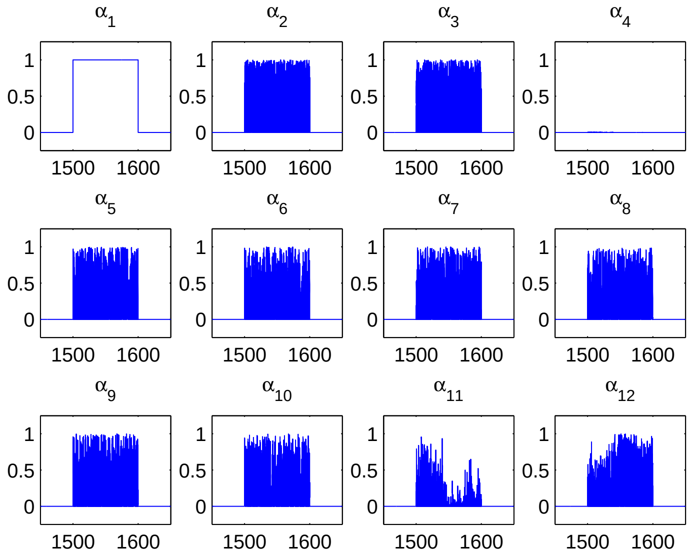

Figure 4: α coeficients in fault scenario f_4

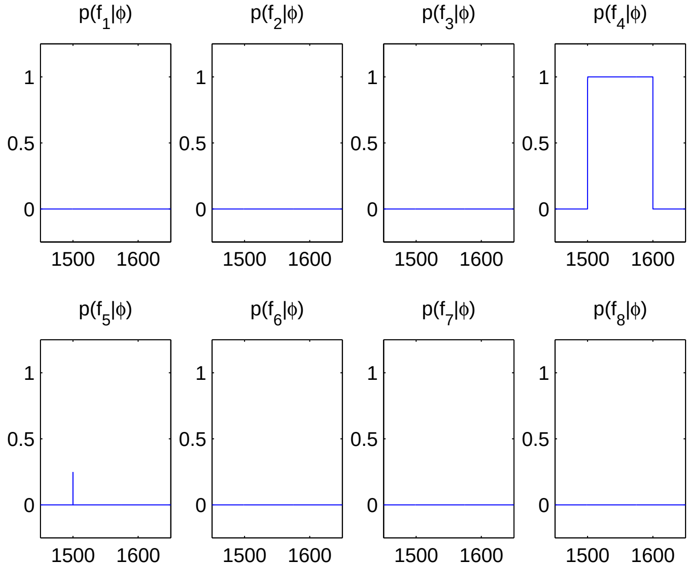

Figure 5: Posterior probabilities in fault scenario f_4

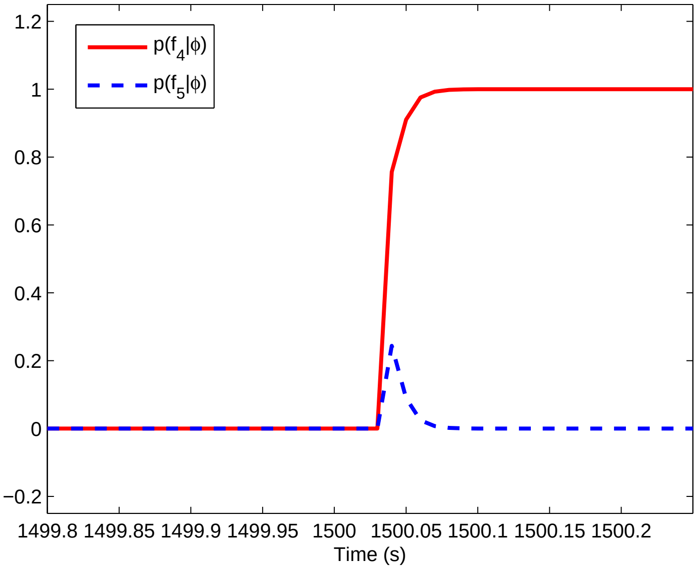

Figure 6: Detail of $p(f_4|\boldsymbol{\phi})$ and $p(f_5|\boldsymbol{\phi})$ in fault scenario f_4

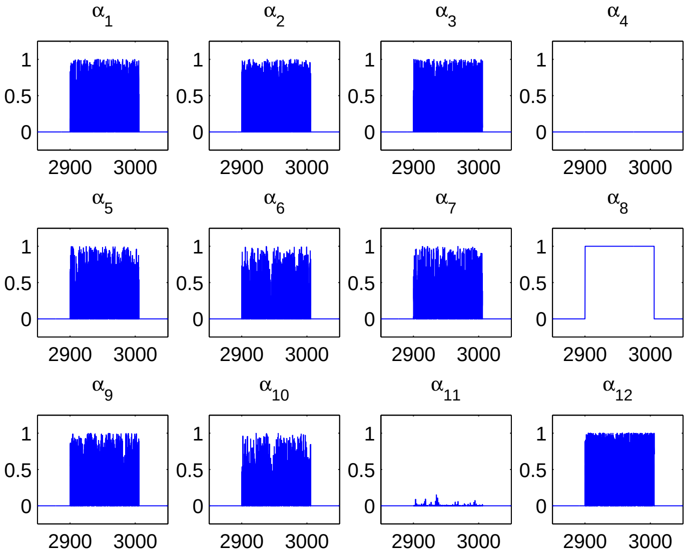

Figure 7: α coeficients in fault scenario f_6

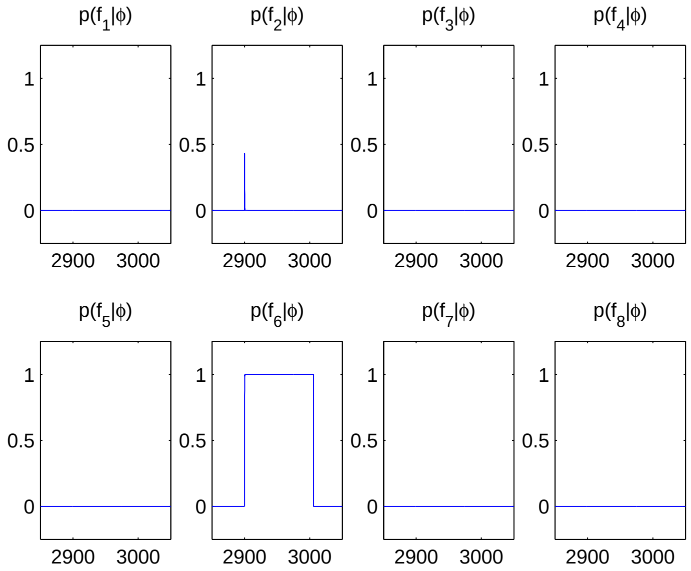

Figure 8: Posterior probabilities in fault scenario f_6

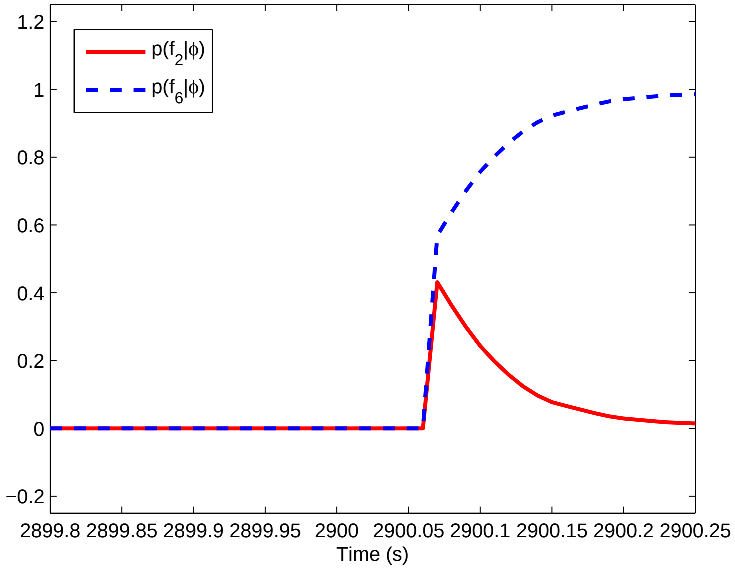

Figure 9: Detail of $p(f_2|\boldsymbol{\phi})$ and $p(f_6|\boldsymbol{\phi})$ in fault scenario f_6

the evolution of the fault posterior probabilities $p(f_i|\boldsymbol{\phi})$, $i = 1, \ldots, 12$, provided by Algorithm 2. As it can be seen in Figure 11 only the posterior probabilities $p(f_3|\boldsymbol{\phi})$ and $p(f_7|\boldsymbol{\phi})$ are different from zero. The detail in Figure 9 shows that, the fault is not permanently correctly isolated $p(f_7|\boldsymbol{\phi}) > p(f_3|\boldsymbol{\phi})$ until the time instant $t = 3411.98s$. In the time interval $t \in [3411.60s, 3411.98s]$, the Algorithm 2 provides either fault f_3 or f_7 as the most probable faults. Once this transient has ended, the fault f_7 is correctly and permanently isolated.

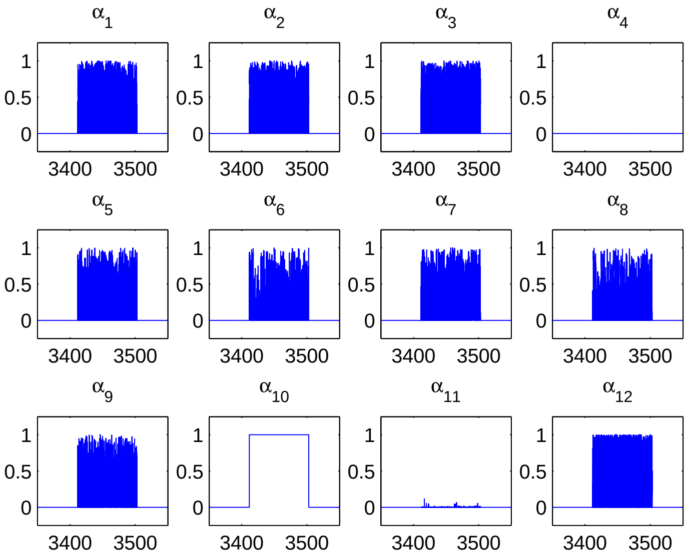

Figure 10: α coeficients in fault scenario f_7

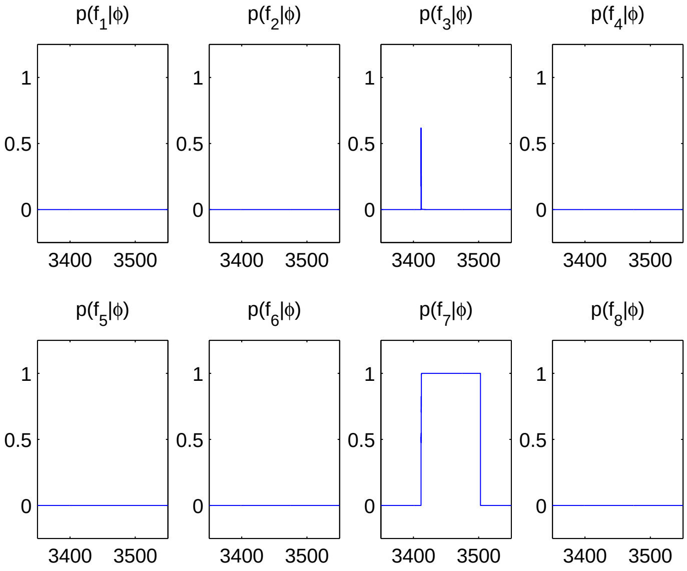

Figure 11: Posterior probabilities in fault scenario f_7

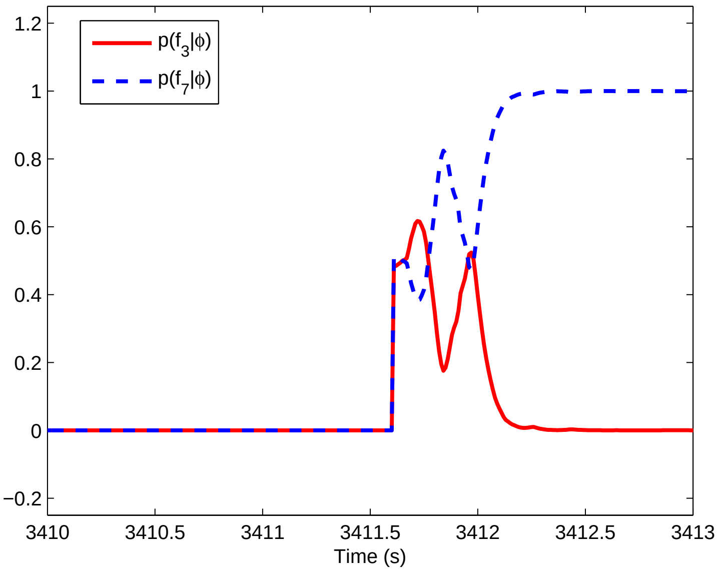

Figure 12: Detail of $p(f_3|\boldsymbol{\phi})$ and $p(f_7|\boldsymbol{\phi})$ in fault scenario f_7

Results summary (for all faults)

In order to verify the performance of the proposed fault detection and isolation method, it has been applied to the whole sequence of eight faults initially considered in the Wind Turbine FDI/FTC benchmark and also presented in Table 1. In fact, as done in the paper [12] that summarizes and compares the successful application of other FDI methods to the benchmark, seven different scenarios have been considered as the result of time-shifting (+100, 0, -100, -200, -300, -400, and -500 s) the occurrence of the faults with respect with the times specified in Table 1 (without changing the wind speed sequence). Moreover, to test the method under different noise realizations, 10 different simulations have been run for each of the seven scenarios. In total, 70 Monte Carlo simulations have been run for each fault, which assures the representativeness of the results.

The obtained results are summarized in Table 4. Results related to the performance of both fault detection and fault isolation tasks are provided. In particular, the minimum ($T_{\underline{d/i}}$), maximum ($T_{\overline{d/i}}$) and mean ($\overline{T}_{d/i}$) values for the detection and isolation times are specified.

Table 4: Summarizing FDI results.

		Faults							
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Detection results	$\overline{T}_d$	0.03s	15s	0.03s	0.03s	0.03	0.06s	11.7s	0.04s
	$T_{\underline{d}}$	0.03s	0.08s	0.03s	0.03s	0.03	0.05s	7.83s	0.04s
	$T_{\bar{d}}$	0.03s	77.1s	0.03s	0.03s	0.04	0.11s	19.1s	0.05s
Isolation results	$\overline{T}_i$	0.03s	15s	0.03s	0.03s	0.03	0.25s	11.9s	0.04s
	$T_{\underline{i}}$	0.03s	0.08s	0.03s	0.03s	0.03	0.05s	7.88s	0.04s
	$T_{\bar{i}}$	0.03s	77.1s	0.03s	0.03s	0.04	1.5s	20s	0.05s

The obtained results can be compared with the ones reported in [12] (detailed in Tables VIII, IX and X of that reference), provided by five different methods: the Gaussian Kernel Support Vector Machine Solution (GKSV); the Estimation-Based Solution (EB); the Up-Down Counter Solution (UDC); the Combined Observer and Kalman Filter Solution (COK); and, finally, the General Fault Model Solution (GFM).

A general advantage of the proposed method is that false alarms are eliminated while all the faults are detected (after some detection time) in all the test scenarios (in comparison, all the methods but the GKSV provide some false alarms in some scenarios; and for all the methods some faults remain undetected in some scenarios). In a more detailed analysis, the results obtained for each particular fault can be compared.

For fault f_1, the proposed method and also the GKSV, EB, UDC and GFM methods eliminate the false alarms while providing small detection times ($\overline{T}_d$=0.02s for GKSV and EB; $\overline{T}_d$=0.03s for UDC and the proposed method; $\overline{T}_d$=0.04s for GFM); on the other hand, the COK method provides some false alarms and moreover a much higher mean detection time ($\overline{T}_d$=10.32s).

For fault f_2, the GFM method is the fastest in terms of mean detection time but closely followed by the proposed method ($\overline{T}_d$=13.7s vs. $\overline{T}_d$=15s), and it must be taken into account the proposed method eliminates false alarms while GFM do not; the COK method provides some false alarms and the detection time is a little bit higher ($\overline{T}_d$=29.24s); all the other methods provide much higher detection times ($\overline{T}_d$=47.24s for GKSV; $\overline{T}_d$=44.65s for EB; $\overline{T}_d$=69.12s for UDC) and moreover the EB method produces some false alarms.

For fault f_3, the proposed method and both the GKSV and UDC methods eliminate the false alarms while providing small detection times ($\overline{T}_d$=0.02s for GKSV; $\overline{T}_d$=0.03s for the proposed method; $\overline{T}_d$=0.04s for UDC); all the other methods generate false alarms and moreover provide higher detection times.

For fault f_4, the proposed method is almost as fast as the UDC ($\overline{T}_d$=0.02s vs. $\overline{T}_d$=0.03s) but the latter generates some false alarms; compared with GKSV and EB, both methods also eliminates false alarms but provide higher detection times ($\overline{T}_d$=0.11s and $\overline{T}_d$=0.33s, respectively); finally, the COK and GFM methods generate false alarms and provide higher detection times.

For fault f_5, the proposed method provides much smaller detection times ($\overline{T}_d$=0.03s vs. $\overline{T}_d$=25.9s) than the GKSV that is the only one that eliminates false alarms; on the other hand, the EB method is a little bit faster ($\overline{T}_d$=0.01s vs. $\overline{T}_d$=0.03s) but it generates too many false alarms; all the other methods generate false alarms and provide higher detection times.

For fault f_6, the proposed method clearly provides much better results than all the other methods: the GKSV method do not even detect the fault, the other methods generate false alarms and provide detection times two orders of magnitude higher.

For fault f_7, the only method that is comparable to the one proposed in this paper is the UDC, although generates some false alarms and provides a slightly worse detection time ($\overline{T}_d$=12.93s vs. $\overline{T}_d$=11.7s); all the other methods provide worse results: some false alarms and around two times the detection time.

Finally, for fault f_8, the proposed method provides good results in all the scenarios while the other methods are only able to perform well for the original fault sequence specified in Table 1 (which they are designed on).

In summary, all the methods compared in [12] provide good results for some faults and bad results for others, being the proposed method unique in the sense that provides good results (comparable or better than the obtained by the best of the other methods for the particular fault) for all faults and scenarios.

6. Conclusion

This paper has proposed a fault detection and isolation procedure in a mixed Bayesian/set-membership framework where a combined statistical/set-membership description of the noise is proposed. In particular, the noise is assumed to be unknown but bounded following the set-membership description, but inside the bounds the noise follows a statistical distribution. The fault detection stage produces not only a fault indicator for the cases fault and no-fault but also a consistency index indicating the certainty that the no-fault observation does not correspond to a missing fault. The fault indicator and consistency indexes, along with the structural information about the fault stored in the theoretical fault signature matrix, are used in the fault isolation stage, where a new fault isolation scheme that is inspired in the Bayesian fault isolation framework is developed. The main difference with respect to the existing approaches is that in our approach only models of nominal no faulty behavior are needed. The presented FDI method has been satisfactorily assessed against the wind turbine FDI benchmark proposed in the literature where a set of typical fault scenarios have been developed. As a future work, the authors propose to extend the proposed approach to deal with outliers and fault models by further exploiting the Bayesian framework.

References

[1] E. A. Bossanyi, Wind turbine control for load reduction, Wind Energy 6 (2003) 229–244.

[2] S. Simani, P. Castaldi, Active actuator fault-tolerant control of a wind turbine benchmark model, International Journal of Robust and Nonlinear Control (in press).

[3] S. M. Tabatabaeipour, P. F. Odgaard, T. Bak, J. Stoustrup, Fault Detection of Wind Turbines with Uncertain Parameters: A Set-Membership Approach, Energies 5 (7) (2012) 2424–2448.

[4] C. Sloth, T. Esbensen, J. Stoustrup, Active and passive fault-tolerant LPV control of wind turbines, in: Proceedings of the American Control Conference, Baltimore, USA, 2010, pp. 4640–4646.

[5] R. Isermann, Fault Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance, Springer,New York, 2006.

[6] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, Diagnosis and Fault-Tolerant Control, Springer-Verlag Berlin Heidelberg, 2006.

[7] J. Gertler, Fault Detection and Diagnosis in Engineering Systems, Marcel Dekker, New York, 1998.

[8] J. Chen, R. J. Patton, Robust Model-based Fault Diagnosis for Dynamic Systems, Kluwer Academic Publishers, 1999.

[9] X. Wei, M. Verhaegen, T. van den Engelen, Sensor fault diagnosis of wind turbines for fault tolerant, in: Proceedings of 17th IFAC World Congress, Seoul, Korea, 2008.

[10] P. Odgaard, J. Stoustrup, R. Nielsen, C. Damgaard, Observer based detection of sensor faults in wind turbines, in: Proceedings of the European Wind Energy Conference, 2009.

[11] C. Dobrila, R. Stefansen, Fault tolerant wind turbine control, Master's thesis, Technical University of Denmark, Kgl. Lyngby, Denmark. (2007).

[12] P. Odgaard, J. Stoustrup, M. Kinnaert, Fault-tolerant control of wind turbines: A benchmark model, IEEE Transactions on Control Systems Technology 21 (4) (2013) 1168–1182.

[13] P. F. Odgaard, J. Stoustrup, Results of a Wind Turbine FDI Competition, in: 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS), Mexico City, Mexico, 2012, pp. 102–107.

[14] P. Odgaard, J. Stoustrup, An evaluation of fault tolerant wind turbine control schemes applied to a benchmark model, in: 2014 IEEE Conference on Control Applications, CCA 2014, 2014, pp. 1366–1371.

[15] J. Dong, M. Verhaegen, Data driven fault detection and isolation of a wind turbine benchmark, in: Preprints of the 18th IFAC World Congress, Milan, Italy, 2011, pp. 7086–7091.

[16] N. Laouti, N. Sheibat-Othman, S. Othman, Support vector machines for fault detection in wind turbines., in: Proceedings of 18th IFAC World Congress, Milan, Italy, 2011, pp. 7067–7072.

[17] X. Zhang, Q. Zhang, S. Zhao, R. Ferrari, M. Polycarpou, T. Parisini, Fault detection and isolation of the wind turbine benchmark: An estimation-based approach, in: Proceedings of 18th IFAC World Congress, Vol. 2, 2011, pp. 8295–8300.

[18] A. Ozdemir, P. Seiler, G. Balas, Wind turbine fault detection using counter-based residual thresholding., in: Proceedings of 18th IFAC World Congress, Milan, Italy, 2011, pp. 8289–8294.

[19] W. Chen, S. Ding, A. Sari, A. Naik, A. Khan, S. Yin, Observer-based fdi schemes for wind turbine benchmark, in: Proceedings of 18th IFAC World Congress, Milan, Italy, 2011, pp. 7073–7078.

[20] C. Svard, M. Nyberg, Automated design of an fdi system for the wind turbine benchmark., Proceedings of IFAC World Congress (2011) 8307–8315.

[21] M. Sami, R. J. Patton, An ftc approach to wind turbine power maximisation via t-s fuzzy modelling and control, in: 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Mexico City, Mexico, 2012, pp. 349–354.

[22] S. Simani, P. Castaldi, Adaptive fault-tolerant control design approach for a wind turbine benchmark, in: 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS), Mexico City, Mexico, 2012, pp. 319–324.

[23] J. Blesa, V. Puig, J. Romera, J. Saludes, Fault diagnosis of wind turbines using a set-membership approach., in: Proceedings of the 18th IFAC World Congress, Milan, Italy, 2011, pp. 8316–8321.

[24] D. Rotondo, F. Nejjari, V. Puig, J. Blesa, Fault Tolerant Control of the Wind Turbine Benchmark Using Virtual Sensors/Actuators , in: Proceedings of the 8th IFAC International Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS2012), Mexico City, Mexico, 2012.

[25] M. Basseville, I. Nikiforov, Detection of Abrupt Changes: Theory and Applications, Prentice Hall, 1993.

[26] M. Nyberg, M. Krysander, Combining ai,fdi and statistical hypothesis-testing in a framework for diagnosis, in: Proceedings of the IFAC SAFEPROCESS, washington DC (USA), 2003.

[27] M. Pernestål, B. Wahlberg, M. Nyberg, A Bayesian approach to fault isolation - structure estimation and inference, in: IFAC Safeprocess'06, Beijing, China., 2006.

[28] M. Nyberg, A General Framework for Model Based Diagnosis Based on Statistical Hypothesis, in: 12th International Workshop on Principles of Diagnosis (DX'01), Sansicario, Via Lattea, Italy, 2001, pp. 135–142.

[29] V. Puig, J. Quevedo, T. Escobet, F. Nejjari, S. de las Heras, Passive Robust Fault Detection of Dynamic Processes using Interval Models, IEEE Transactions on Control Systems Technology 16 (2008) 1083–1089.

[30] V. Puig, Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies, International Journal of Applied Mathematics and Computer Science 20 (4) (2010) 619–635.

[31] H. E. Merritt, Hydraulic control systems, John Wiley and Sons, Inc., 1967.

[32] C. Sloth, T. Esbensen, J. Stoustrup, Robust and Fault-Tolerant Linear Parameter-Varying Control of Wind Turbines, Mechatronics 21 (4) (2011) 645–659.

[33] R. M. Fernandez, S. Tornil-Sin, J. Blesa, V. Puig, Non-linear set-membership identification approach based on the bayesian framework, IET Control Theory and Applications 9 (9) (2015) 1392–1398.

[34] H. Sorenson, Least-squares estimation: from gauss to kalman, IEEE Spectrum 7 (7) (1970) 63–68.

[35] V. Puig, S. Montes de Oca, J. Blesa, Adaptive threshold generation in robust fault detection using interval models: time-domain and frequency-domain approaches, International Journal of Adaptive Control and Signal Processing 27 (10) (2013) 873–901.

[36] J. Blesa, D. Rotondo, V. Puig, F. Nejjari, FDI and FTC of wind turbines using the interval observer approach and virtual actuators/sensors, Control Engineering Practice 24 (2014) 138 – 155.

[37] M. Blanke, T. Lorentz, Satool: A software tool for structural analysis of complex automation systems, in: 6th IFAC SAFEPROCESS, 2006, pp. 673–678.

[38] P. F. Odgaard, J. Stoustrup, M. Kinnaert, Fault-tolerant control of wind turbines: a benchmark model, IEEE Transactions on Control Systems Technology 21 (4) (2013) 1168–1182.

[39] C. Svärd, M. Nyberg, Automated design of an fdi system for the wind turbine benchmark, Journal of Control Science and Engineering (2012) 1–13.