

# FDI and FTC of Wind Turbines using the Interval Observer Approach and Virtual Actuators/Sensors

Joaquim Blesa<sup>a,b</sup>, Damiano Rotondo<sup>a</sup>, Vicenç Puig<sup>a,b</sup>, Fatiha Nejjari<sup>a</sup>

<sup>a</sup>*Automatic Control Department, Universitat Politècnica de Catalunya (UPC), TR11, Rambla de Sant Nebridi, 10, 08222, Terrassa, Spain.*

<sup>b</sup>*Institut de Robòtica i Informàtica Industrial (IRI), UPC-CSIC, Llorens i Artigas, 4-6, 08028 Barcelona, Spain.*

---

## Abstract

In this work, the problem of Fault Detection and Isolation (FDI) and Fault Tolerant Control (FTC) of wind turbines is addressed. Fault detection is based on the use of interval observers and unknown but bounded description of the noise and modeling errors. Fault isolation is based on analyzing the observed fault signatures on-line and matching them with the theoretical ones obtained using structural analysis and a row-reasoning scheme. Fault tolerant control is based on the use of virtual sensors/actuators to deal with sensor and actuator faults, respectively. More precisely, these FTC schemes, that have been proposed previously in state space form, are reformulated in input/output form. Since an active FTC strategy is used, the FTC module uses the information from the FDI module to replace the real faulty sensor/actuator by activating the corresponding virtual sensor/actuator. Virtual actuators/sensors require additionally a fault estimation module to compensate the fault. In this work, a fault estimation approach based on batch least squares is used. The performance of the proposed FDI and FTC schemes is assessed using the proposed fault scenarios considered in the wind turbine benchmark proposed at IFAC SAFEPROCESS 2009. Satisfactory results have been obtained in both FDI and FTC.

*Keywords:* Fault detection, Uncertainty, Interval observer, Wind turbine, Fault tolerant control, Virtual sensors, Virtual actuators.

---

\*Corresponding author: Joaquim Blesa, Department of Automatic Control (ESAI), Technical University of Catalonia (UPC), Rambla de Sant Nebridi 10, 08222 - Terrassa (Spain). Tel: +34 93 739 85 63 E-mail address: joaquim.blesa@upc.edu

## 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 non-planned 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 (Bossanyi, 2003; Simani & Castaldi, 2013). Hence, the detailed modeling of wind turbines has been a hot topic of research in the last years (van der Veen et al., 2013). Another way of reducing the costs is developing wind turbines that require less scheduled and especially non-scheduled service and have less downtime due to failure (Tabatabaeipour et al., 2012). 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 (Sloth et al., 2010). 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 (Isermann, 2006). Model-based FDI is often necessary to obtain a good diagnosis of faults.

The problem of model-based fault diagnosis in wind turbines has recently been addressed (Odgaard & Stoustrup, 2012b), the main motivation being the importance gained in many countries by this technology for electricity generation. So far, revising the literature, methods ranging from Kalman filters (Wei et al., 2008), observers (Odgaard et al., 2009), parity equations (Dobrila & Stefansen, 2007), dynamic weighting ensembles (Razavi-Far & Kinnaert, 2013) and fuzzy modeling and identification methods (Badihi et al., 2013) have already been suggested as possible model-based techniques for fault diagnosis of wind turbines.

The problem of model-based fault tolerant control in wind turbines has been addressed even more recently. In Sloth et al. (2010) and Sloth et al. (2011), active and passive fault tolerant control designs for wind turbines are presented. The Linear Parameter Varying (LPV) control design method is applied, which leads to LMI based optimization in case of active fault tolerant and Bilinear Matrix Inequalities (BMIs) in case of passive fault tolerant problems. It is shown through simulations that both active and passive controllers have better performance than classical PI controller and that active fault-tolerant controller is better than passive FTC in faulty condition. However, the authors conclude that the choice between active and passive FTC should also take into account the tolerance to errors in the fault diagnosis system. In Sami & Patton (2012), a robust FTC strategy that optimizes the wind energy captured by a wind turbine operating at low wind speeds

(5 MW), using an adaptive gain Sliding Mode Control (SMC) is proposed. The proposed method involves a robust descriptor observer design that can provide simultaneously a robust estimation of the states and the "unknown outputs" (sensor faults and noise) in order to guarantee the robustness of the sliding surface against unknown output effects. In Odgaard & Stoustrup (2012a), an FTC scheme based on estimates of the generator speed using a bank of unknown input observers, and considering faults in the rotor and generator speed sensors, is proposed. One observer is designed for each of the sets of non faulty rotor and generator speed sensors. The unknown input observers are used to detect and isolate these faults too. In Kamal et al. (2012), a multiobserver switching control strategy for robust active fault tolerant fuzzy control of variable-speed wind energy conversion systems in the presence of wide wind variation, wind disturbance, parametric uncertainties and sensor faults is proposed. In Badihi et al. (2013), fault tolerance is achieved using a gain-scheduled Proportional Integral control system based on Fuzzy Gain Scheduling. A projection-based approach is used by Jain et al. (2013) in order to obtain an active FTC system that neither uses a priori information about the model of the wind turbine in real-time nor an explicit fault diagnosis scheme. An active FTC scheme based on adaptive filters obtained via the nonlinear geometric approach is proposed in Simani & Castaldi (2013), allowing to obtain an interesting decoupling property with respect to uncertainty affecting the wind turbine system.

The use of on-line fault estimation is essential for all active fault compensation approaches. A number of suitable estimation methods, essentially observer-based or Kalman filter-based fault estimation are proposed in the literature (Wang & Daley, 1996; Edwards et al., 2000; Patton & Klinkhieo, 2009). In Montes de Oca et al. (2011), a recursive least square method is applied for actuator fault estimation in LPV systems.

In Odgaard et al. (2013), a benchmark model for fault detection and isolation as well as fault tolerant control of wind turbines has been proposed. The benchmark model describes a realistic generic three blade horizontal variable speed wind turbine with a full scale converter coupling and a rated power of 4.8MW. Solutions to FDI and FTC for this benchmark model have been published recently: (Chen et al., 2011), (Blesa et al., 2011), (Tabatabaeipour et al., 2012) and (Rotondo et al., 2012), among others, and compared in Odgaard et al. (2013).

In this paper, the problem of fault diagnosis in wind turbines is addressed applying the interval observer based approach proposed in Puig et al. (2006). The proposed model based fault detection methodology relies on the use of interval observers and assumes an unknown but bounded description of the noise and the modeling errors. Fault isolation is based on analyzing the observed fault signa-

tures on-line and matching them with the theoretical ones obtained using structural analysis and a row-reasoning scheme. On the other hand, the fault tolerant control approach considered in this work uses the idea of virtual sensors/actuators. The paper suggests the reformulation of these FTC schemes, previously proposed in state space form by Lunze et al. (2003), in an input/output form. A fault estimation scheme based on batch least squares approach is also suggested. The performance of the proposed FTC schemes is assessed using the fault scenarios considered in the FTC benchmark presented in Odgaard et al. (2013).

In Section 2 the proposed fault detection and isolation based on interval observers is presented. In Section 3, the proposed fault tolerant control approach based on virtual sensors and actuators is introduced. In Section 4, the wind turbine used in the FDI/FTC competition is briefly introduced and the set of residuals generated using structural analysis. Results of the application of the proposed FDI/FTC approaches to the wind turbine benchmark are presented in Section 5. Finally, some conclusions are drawn in Section 6.

## 2. Fault Detection, Isolation and Estimation

### 2.1. Problem set-up

Let us consider that the wind turbine to be monitored can be described by a MIMO linear uncertain dynamic model expressed as follows:

$$x(k+1) = A(\tilde{\theta})x(k) + B(\tilde{\theta})u(k) + F_a(\tilde{\theta})f_a(k) \quad (1)$$

$$y(k) = C(\tilde{\theta})x(k) + F_y(\tilde{\theta})f_y(k) + \tilde{v}(k) \quad (2)$$

where  $u(k) \in \mathbb{R}^{n_u}$  is the system input,  $y(k) \in \mathbb{R}^{n_y}$  is the system output,  $x(k) \in \mathbb{R}^{n_x}$  is the state-space vector,  $\tilde{v}(k) \in \mathbb{R}^{n_y}$  is the output noise that is assumed to be bounded  $|\tilde{v}_i(k)| < \sigma_i$  with  $i = 1, \dots, n_y$ ,  $f_a(k) \in \mathbb{R}^{n_u}$  and  $f_y(k) \in \mathbb{R}^{n_y}$  represent faults in the actuators and output sensors, respectively.  $A(\tilde{\theta})$ ,  $B(\tilde{\theta})$ ,  $C(\tilde{\theta})$ ,  $F_a(\tilde{\theta})$  and  $F_y(\tilde{\theta})$  are matrices of appropriate dimensions where  $\tilde{\theta} \in \mathbb{R}^{n_\theta}$  is the parameter vector.

The system (1)-(2) is monitored using a linear observer with Luenberger structure that uses an *interval model* of the system, i.e., a model with parameters bounded by intervals<sup>1</sup>:

$$\theta \in \Theta = \left\{ \theta \in \mathbb{R}^{n_\theta} \mid \underline{\theta}_i \leq \theta_i \leq \bar{\theta}_i, i = 1, \dots, n_\theta \right\} \quad (3)$$

---

<sup>1</sup>The intervals for uncertain parameters can be inferred from real data using set-membership parameter estimation algorithms (Milanese et al., 1996; Ploix et al., 1999).

that represent the uncertainty about the exact knowledge of the real parameters  $\tilde{\theta}$ . This observer, known as an *interval observer*, is expressed as follows (Meseguer et al., 2010):

$$\begin{aligned}\hat{x}(k+1, \theta) &= (A(\theta) - LC(\theta))\hat{x}(k, \theta) + B(\theta)u(k) + Ly(k) \\ &= A_0(\theta)\hat{x}(k, \theta) + B(\theta)u(k) + Ly(k) \\ \hat{y}(k, \theta) &= C(\theta)\hat{x}(k, \theta)\end{aligned}\quad (4)$$

where  $\hat{x}(k, \theta)$  is the estimated system state vector,  $\hat{y}(k, \theta)$  is the estimated system output vector and  $A_0(\theta) = A(\theta) - LC(\theta)$  is the observer matrix.

The observer gain matrix  $L \in \mathbb{R}^{n_x \times n_y}$  is designed to stabilize the matrix  $A_0(\theta)$  and to guarantee a desired performance regarding fault detection for all  $\theta \in \Theta$  using the LMI pole placement approach (Chilali & Gahinet, 1996).

The input/output form of the system (1)-(2) using the shift operator  $q^{-1}$  and assuming zero initial conditions is given by:

$$y(k) = y_0(k, \tilde{\theta}) + G_{f_a}(q^{-1}, \tilde{\theta})f_a(k) + G_{f_y}(\tilde{\theta})f_y(k) + \tilde{v}(k) \quad (5)$$

where  $y_0(k, \tilde{\theta})$  is the system output when the system is not affected by faults, disturbances and noises:

$$y_0(k, \tilde{\theta}) = G_u(q^{-1}, \tilde{\theta})u(k) \quad (6)$$

$$G_u(q^{-1}, \tilde{\theta}) = C(\tilde{\theta})(qI - A(\tilde{\theta}))^{-1}B(\tilde{\theta}) \quad (7)$$

$$G_{f_a}(q^{-1}, \tilde{\theta}) = C(\tilde{\theta})(qI - A(\tilde{\theta}))^{-1}F_a(\tilde{\theta}) \quad (8)$$

$$G_{f_y}(\tilde{\theta}) = F_y(\tilde{\theta}) \quad (9)$$

The input/output form of the observer (4) is expressed as follows:

$$\hat{y}(k, \theta) = G(q^{-1}, \theta)u(k) + H(q^{-1}, \theta)y(k) \quad (10)$$

with:

$$G(q^{-1}, \theta) = C(\theta)(qI - A_0(\theta))^{-1}B(\theta) \quad (11)$$

$$H(q^{-1}, \theta) = C(\theta)(qI - A_0(\theta))^{-1}L \quad (12)$$

The effect of the uncertain parameters  $\theta$  on the observer temporal response  $\hat{y}(k, \theta)$  will be bounded using an interval satisfying:

$$\hat{y}(k, \theta) \in [\underline{\hat{y}}(k), \bar{\hat{y}}(k)] \quad (13)$$

Such interval can be computed independently for each output  $i = 1, \dots, n_y$ , neglecting couplings among outputs, as follows:

$$\underline{\hat{y}}_i(k) = \min_{\theta \in \Theta} \hat{y}_i(k, \theta) \quad \text{and} \quad \bar{\hat{y}}_i(k) = \max_{\theta \in \Theta} \hat{y}_i(k, \theta) \quad (14)$$

subject to the observer equations given by (4). The optimization problems (14) could be solved using numerical methods as in Puig et al. (2003). However, in this paper, a zonotopic approach (Puig et al., 2013), whose complexity is linear with respect to the system dimension since it involves only matricial operation, and therefore is more efficient from the computational point of view, will be used as described below.

Finally, taking into account that the additive noise in the system (2) is bounded, the following condition should be satisfied in a non-faulty scenario:

$$y_i(k) \in \left[ \underline{\hat{y}}_i(k) - \sigma_i, \bar{\hat{y}}_i(k) + \sigma_i \right] \quad i = 1, \dots, n_y \quad (15)$$

## 2.2. Implementation using zonotopes

In order to compute the interval (13) at present instant from previous intervals determined in previous time instants using zonotopes, the observer (4) can be formulated as follows:

$$\hat{x}(k, \theta) = A_0(\theta)\hat{x}(k-1, \theta) + B_0(\theta)u_0(k-1) \quad (16)$$

$$\hat{y}(k, \theta) = C(\theta)\hat{x}(k, \theta) \quad (17)$$

where:  $A_0(\theta) = A(\theta) - LC(\theta)$ ,  $B_0(\theta) = [B(\theta) \quad L]$  and  $u_0(k) = [u(k) \quad y(k)]^T$ .

**Definition 1.** Given the sequence of measured inputs  $\{u(i)\}_0^{k-1}$  and outputs  $\{y(i)\}_0^{k-1}$  and assuming that the initial states are bounded by a known compact set  $\mathbb{X}_0$ . Then, the exact uncertain estimated state set  $\mathbb{X}_k$  using at time  $k$  (16) is expressed by:

$$\mathbb{X}_k = \{ \hat{x}(k, \theta) : (\hat{x}(i, \theta) = A_0(\theta)\hat{x}(i-1, \theta) + B_0(\theta)u_0(i-1))_{i=1}^k \mid \hat{x}_0 \in \mathbb{X}_0, \theta \in \Theta \} \quad (18)$$

The uncertain state set described in *Definition 1* at time  $k$  can be computed approximately by admitting the rupture of the existing relations between variables of consecutive time instants. This allows to compute an approximation of this set from the approximate uncertain set at time  $k-1$ .

**Definition 2.** Consider the interval observer given by (16), the set of uncertain states at time  $k-1$  ( $\mathbb{X}_{k-1}$ ) and the input/output values  $\{u(k-1), y(k-1)\}$ , then the approximate set of estimated states  $\mathbb{X}_k^e$  at time  $k$  based on the measurements up to time  $k-1$  is defined as:

$$\mathbb{X}_k^e = \{\hat{x}(k, \theta) : \hat{x}(k, \theta) = A_0(\theta)\hat{x}(k-1, \theta) + B_0(\theta)u_0(k-1) \mid \hat{x}(k-1, \theta) \in \mathbb{X}_{k-1}, \theta \in \Theta\} \quad (19)$$

*Definition 1* and *Definition 2* can be easily adapted to describe the exact uncertain estimated output set  $\mathbb{Y}_k$  at time  $k$  and the approximate set of estimated outputs  $\mathbb{Y}_k^e$ .

Since the exact set of estimated states  $\mathbb{X}_k^e$  and outputs  $\mathbb{Y}_k^e$  are difficult to compute, one way is to bound them using some zonotopes as in Alamo et al. (2005).

Here, the set of estimated states  $\mathbb{X}_k^e$  (or outputs  $\mathbb{Y}_k^e$ ) introduced in *Definition 2* will be approximated iteratively using zonotopes. From these zonotopes, an interval for each state variable and output can also be obtained by computing the interval hull of the zonotope  $\mathbb{Z}$ , denoted as  $\square\mathbb{Z}$ . The sequence of interval hulls  $\square\mathbb{X}_k^e$  and  $\square\mathbb{Y}_k^e$  with  $k \in [0, N]$  will be called the *interval observer estimation* of the system (16)-(17) where  $N$  is the number of measurement data considered. Following the previous idea, *Algorithm 1* is proposed to determine an approximation of the set of uncertain estimated states and outputs.

---

**Algorithm 1** Interval Observer using Set Computations

---

- 1:  $k \leftarrow 0$
  - 2:  $\mathbb{X}_k^e \leftarrow \mathbb{X}_0$
  - 3: **while**  $k < N$  **do**
  - 4:   Obtain and store input-output data  $\{u(k), y(k)\}$
  - 5:   Compute the approximated estimated state set,  $\mathbb{X}_{k+1}^e$
  - 6:   Compute the approximated estimated output set,  $\mathbb{Y}_k^e$
  - 7:   Compute the interval hull of the approximated estimated output set,  $\square\mathbb{Y}_k^e = \left[ \underline{\hat{y}}(k), \bar{\hat{y}}(k) \right]$
  - 8:    $\bar{k} \leftarrow k + 1$
  - 9: **end while**
- 

The implementation of steps 5-8 in the *Algorithm 1* using zonotopes is described in detail in Puig et al. (2013).

### 2.3. Parameter uncertainty estimation

One of the key points in passive robust model based fault detection is how models and their uncertainty bounds are obtained. Classical system identification methods (Ljung, 1987) are formulated under a statistical framework. Assuming that the measured variables are corrupted by additive noises with known statistical distributions and that the model structure is known, a parameter estimation algorithm will provide nominal values for the parameters together with descriptions of the associated uncertainty in terms of the covariance matrix or confidence regions for a given probability level (Kendall & Stuart, 1979; Dalai et al., 2005). However, this type of approaches cannot be applied when measurement errors are described as unknown but bounded values and/or modeling errors exist. Recently, some methodologies that provide a model with its uncertainty have been developed for control applications (Reinelt et al., 2002). One of the methodologies assumes the bounded but unknown description of the noise and parametric uncertainty. This methodology is known as *bounded-error* or *set-membership estimation* (Milanese et al., 1996), which produces a set of parameters consistent with the selected model structure and the pre-specified noise bounds. This approach is used for estimating parametric uncertainty of the interval observers in (4).

Regarding the uncertain variables in (4), it is assumed that a priori theoretical or practical considerations allow to obtain useful intervals associated to measurement noises, leading to an estimation of the noise bound  $\sigma$ . The goal of the parameter estimation algorithm is to characterize the parameter set  $\Theta$  (here a box) consistent with the data collected in a fault-free scenario. Given  $N$  measurements of system inputs  $y(k)$  and outputs  $u(k)$  from a scenario free of faults and rich enough from the identifiability point of view, and a nominal model described by a vector  $\theta_n$  obtained using least-squares parameter estimation algorithm (Ljung, 1987), the uncertain parameter estimation algorithm proceeds by solving the following optimization problem:

$$\begin{aligned}
& \min \alpha \\
& \text{subject to :} \\
& y_i(k) \in [\underline{\hat{y}}_i(k) - \sigma_i, \bar{\hat{y}}_i(k) + \sigma_i] \quad i = 1, \dots, n_y \quad k = 1, \dots, N \\
& \underline{\hat{y}}_i(k) = \min_{\theta \in \Theta} \hat{y}_i(k, \theta) \quad i = 1, \dots, n_y \quad k = 1, \dots, N \\
& \bar{\hat{y}}_i(k) = \max_{\theta \in \Theta} \hat{y}_i(k, \theta) \quad i = 1, \dots, n_y \quad k = 1, \dots, N \\
& \hat{y}(k, \theta) = G(q^{-1}, \theta)u(k) + H(q^{-1}, \theta)y(k) \quad k = 1, \dots, N \\
& \Theta = [\theta_n(1 - \alpha), \theta_n(1 + \alpha)]
\end{aligned} \tag{20}$$



#### 2.4. Fault detection test

Fault detection is based on generating a nominal residual comparing the measurements of physical system variables  $y(k)$  with their estimation  $\hat{y}(k)$  provided by the observer (4):

$$r^o(k) = y(k) - \hat{y}(k, \theta_n) \quad (21)$$

where  $r(k) \in \mathbb{R}^{n_y}$  is the residual set and  $\theta_n$  the nominal parameters.

According to Gertler (1998), the computational form of the nominal residual generator, obtained using (4), is:

$$r^o(k) = (I - H(q^{-1}, \theta_n))y(k) - G(q^{-1}, \theta_n)u(k) \quad (22)$$

that has been derived taking into account the input/output form of the observer (10).

When considering model uncertainty located in parameters, the residual generated by (21) will not be zero, even in a non-faulty scenario. To cope with the parameter uncertainty effect, a passive robust approach based on adaptive thresholding can be used (Puig et al., 2006). Thus, using this passive approach, the effect of parameter uncertainty in the components  $r_i(k)$  of residual  $r(k)$  (associated to each system output  $y_i(k)$ ) is bounded by the interval (Puig et al., 2003):

$$r_i^o(k) \in [\underline{r}_i(k) - \sigma_i, \bar{r}_i(k) + \sigma_i] \quad i = 1, \dots, n_y \quad (23)$$

where:

$$\underline{r}_i(k) = \underline{\hat{y}}_i(k) - \hat{y}_i(k, \theta_n) \text{ and } \bar{r}_i(k) = \bar{\hat{y}}_i(k) - \hat{y}_i(k, \theta_n) \quad (24)$$

where  $\underline{\hat{y}}_i(k)$  and  $\bar{\hat{y}}_i(k)$  are the bounds of the system output estimation computed component-wise using the interval observer (4) and obtained according to (14).

Then, the fault detection test could be based on checking if the residuals satisfy or not the condition given by (23). In case that this condition does not hold, a fault can be indicated.

**Remark 2.1.** As discussed in Meseguer et al. (2010), fault detection based on interval observers may suffer from missed detection because of the uncertainty. This is due to the fact that there exists a minimum fault size that guarantees the activation of the fault detection test (23) despite the uncertainties. The minimum fault size depends on the uncertainty bounds in such a way that the performance of the fault detection test will decrease when those bounds increase. On the other hand, interval observers guarantee that there are no false alarms since uncertainty bounds are determined to explain the data collected in non-faulty scenarios, as described previously.

## 2.5. Fault isolation

Fault isolation consists in identifying the faults affecting the system. It is carried out on the basis of fault signatures, generated by the detection module, and their relation with all the considered faults,  $f(k) = \{f_a(k), f_y(k)\}$ . Robust residual evaluation presented in Section 2.4 allows obtaining a set of *fault signatures*  $\phi(k) = [\phi_1(k), \phi_2(k), \dots, \phi_{ny}(k)]$ , where each fault indicator is given by:

$$\phi_i(k) = \begin{cases} 0 & \text{if } r_i^o(k) \in [r_i(k) - \sigma_i, \bar{r}_i(k) + \sigma_i] \\ 1 & \text{if } r_i^o(k) \notin [r_i(k) - \sigma_i, \bar{r}_i(k) + \sigma_i] \end{cases} \quad (25)$$

Standard fault isolation reasoning exploits the knowledge about the binary relation between the set of fault hypothesis and the set of residuals that is stored in the so called *Fault Signature Matrix* (FSM), denoted as  $M$ . An element  $m_{i,j}$  of  $M$  is equal to 1 if the fault  $f^j$  affects the computation of the residual  $r_i$ ; otherwise  $m_{i,j} = 0$ . A column of  $M$  is known as a *theoretical fault signature* and indicates which residuals are affected by a given fault. A set of faults is *isolable* if all the columns in  $M$  are different (two identical columns indicate two indistinguishable faults).

Based on the use of FSMs, different reasoning procedures have been proposed in the literature (Cordier et al., 2004). The most common one involves finding a matching between the observed fault signature and one of the theoretical fault signatures. However, this reasoning is not appropriate in an unknown but bounded context. Due to the uncertainty, when a fault is present in the system, an undefined number of the residuals affected by the fault can be found inconsistent, mainly depending on the sensitivity of each residual to the fault and on the fault magnitude. In other words, the observed fault signature will not exactly match the theoretical signature of the present fault. In this case, if the column-matching procedure is used, then the particular fault will not be identified. An appropriate reasoning should only consider the residuals that are inconsistent when searching for the fault, since consistency is not relevant. A residual that is found inconsistent indicates that one of the faults that affect the residual is acting on the system. But the contrary is not true, if a residual is satisfied, it does not assure that none of the associated faults is present. According to the established terminology (Cordier et al., 2004), the used algorithm must avoid *single-fault exoneration*, which is implicit in the column matching reasoning.

Under single-fault assumption, this can be easily achieved by taking into account that the fault that is actually present in the system has to affect all the residuals that have been found inconsistent according to the observed fault signature (if

not, the single fault hypothesis can not explain the observed behavior). Algorithm 2 summarizes an isolation procedure based on this idea.

**Remark 2.2.** Due to the uncertainty, it is possible that the observed fault signature may be attributed to more than one fault and hence more than one fault candidate is provided by Algorithm 2. The most probable candidate is selected at the end taking into account the number of activated residuals with respect to the expected ones. On the other hand, it can always be assured that the real fault present in the system is one of the proposed fault candidates.

### 2.6. Fault estimation

In this paper, fault estimation is formulated as a parameter estimation problem in such a way that any parameter estimation algorithm can be used. In this paper, the *sliding window blockwise least-squares* method (Jiang & Zhang, 2004) is used.

In order to apply the block least-squares, the model including faults must be rewritten in a linear regression form:

$$z(k) = \varphi(k)\vartheta \quad (26)$$

where  $z(k)$  and  $\varphi(k)$  are signals that are either directly measured or obtained using some mathematical or physical relationship between the measured variables, and  $\vartheta$  is the parameter to be estimated, that is, the fault.

Taking into account the last  $N$  samples (26) can be written as:

$$Z(k) = \Phi(k)\vartheta \quad (27)$$

with:

$$Z(k) = \begin{bmatrix} z(k) \\ z(k-1) \\ \vdots \\ z(k-N+1) \end{bmatrix} \quad \Phi(k) = \begin{bmatrix} \varphi(k) \\ \varphi(k-1) \\ \vdots \\ \varphi(k-N+1) \end{bmatrix} \quad (28)$$

the fault estimation can be obtained by:

$$\hat{\vartheta} = \Phi^\dagger(k)Z(k) \quad (29)$$

where  $\Phi^\dagger(k)$  denotes the pseudoinverse of  $\Phi(k)$ .

Notice that the design parameter in this algorithm is the length of the rectangular sliding window  $N$ . The choice of this parameter should be made taking into account that small values of  $N$  lead to fault estimations that are faster but more sensitive to noise, while with big values of  $N$  the obtained fault estimation is slower but more robust against noise.

---

**Algorithm 2** Fault detection and isolation

---

```
1:  $fault \leftarrow \text{FALSE}$ 
2:  $k \leftarrow 0$ 
3: while  $fault \neq \text{TRUE}$  do
4:    $k \leftarrow k + 1$ 
5:   Obtain input-output data  $\{u(k), y(k)\}$  at time instant  $k$ 
6:   Compute  $[y_i(k), \bar{y}_i(k)], i = 1, \dots, n_y$  using Algorithm 1
7:   Obtain  $[r_i(\bar{k}), \bar{r}_i(k)], i = 1, \dots, n_y$  using Eq. (24)
8:   for  $i = 1$  to  $n_y$  do
9:     if  $r_i^o(k) \notin [r_i(k) - \sigma_i, \bar{r}_i(k) + \sigma_i]$  then
10:       $\phi_{ik} = 1$ 
11:       $fault \leftarrow \text{TRUE}$ 
12:     else
13:       $\phi_{ik} = 0$ 
14:     end if
15:   end for
16:    $\mathcal{FC} \leftarrow \{f^1, f^2, \dots, f^{n_f}\}$ 
17:   for  $i = 1$  to  $n_y$  do
18:     if  $\phi_{ik} = 1$  then
19:       for  $j = 1$  to  $n_f$  do
20:         if  $m_{i,j} = 0$  then
21:            $\mathcal{FC} \leftarrow \mathcal{FC} - f^j$ 
22:         end if
23:       end for
24:     end if
25:   end for
26: end while
27: Fault candidate set  $\mathcal{FC}$ 
28: for  $j = 1$  to  $|\mathcal{FC}|$  do
29:    $\pi_j(k) = \frac{\sum_{i=1}^{n_y} \phi_i(k) m_{i,j}}{\sum_{i=1}^{n_y} m_{i,j}}$ 
30: end for
31: Candidate fault is:  $\arg \max_{j=1, \dots, |\mathcal{FC}|} \pi_j(k)$ 
```

---

### 3. Fault Tolerant Control

When designing fault-tolerant control systems, fault tolerance should be addressed either on the sensors or the actuators of the system. Fault-tolerance meth-

ods generally assume redundancy, that is, the existence of redundant actuators or sensors that can be used in faulty situations. In the case of actuators/sensors, there are two ways of including fault-tolerance in the control loop: namely, by introducing auxiliary actuators/sensors (hardware redundancy) that will replace the faulty ones, or by using the existing mathematical relationships that describe the system behavior, in order to compensate the faults (analytical redundancy).

### 3.1. Hardware redundancy

Hardware redundancy is the most used way of introducing redundancy in actuators/sensors in industry (see Fig. 1). In real time, an FDI module checks if the operating actuator/sensor that is operating is working properly or not. In case it is not, such an actuator/sensor is disconnected and replaced by the redundant one. This reconfiguration mechanism hides the fault from the controller side and there is no need of retuning the controller parameters. However, this solution is in general costly from the economical point of view.

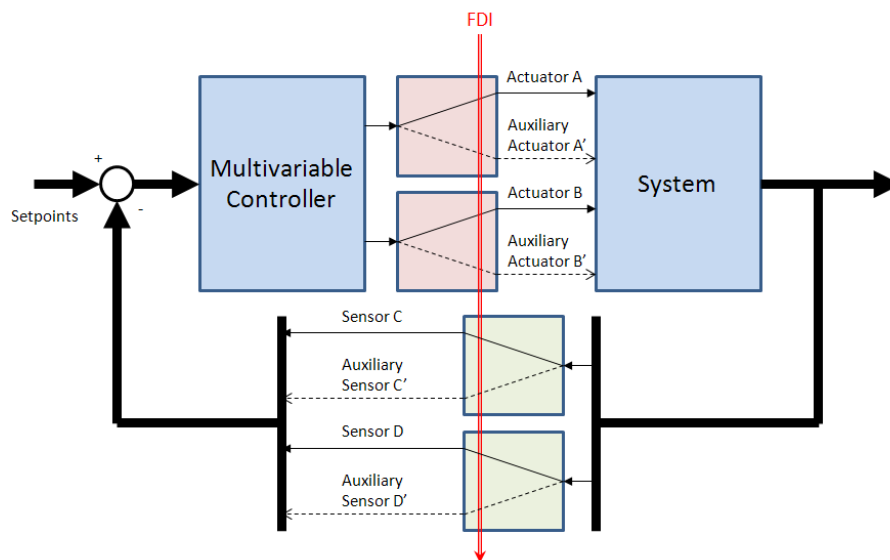


Figure 1: Hardware redundancy.

### 3.2. Analytical redundancy

On the other hand, analytical redundancy tries to exploit the redundancy already existing in the system through the use of models, in order to correct the

closed-loop behavior in such a way that it behaves as in the non-faulty case. There are two ways to achieve this goal: (a) either by retuning the controller parameters such that the performance of the closed loop in faulty situation tries to be as close as possible to the one obtained in non-faulty situation. This method is known as *model matching approach* (Blanke et al., 2006); (b) or by designing a virtual actuator  $G_a(q^{-1})$  or virtual sensor  $G_s(q^{-1})$  block that is placed between the process and the controller or between the sensors and the controller, respectively, allowing to hide the fault with respect to the controller, that in this case should not be retuned (Lunze & Steffen, 2003). In this paper, this second methodology is extended to the input/output formulation and is used to achieve fault tolerance.

Given a multivariable system with  $m$  inputs and  $n$  outputs described by a transfer matrix  $G(q^{-1})$ , and controlled by a controller  $K(q^{-1})$ , in case that a fault appears in the system changing it to  $G_f(q^{-1})$ , the virtual actuator  $G_a(q^{-1})$  is placed between the process and the controller (see Fig. 2) and is designed to satisfy:

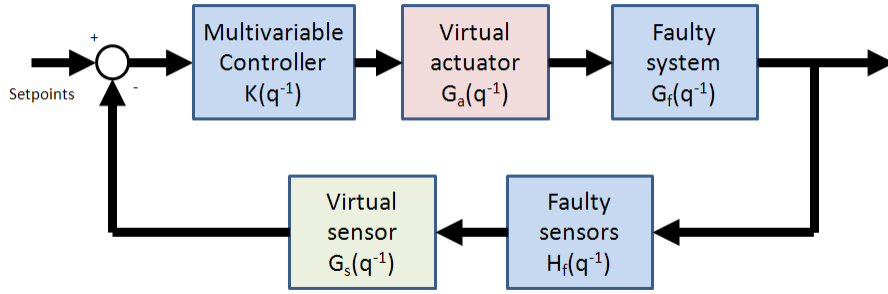


Figure 2: Analytical redundancy.

$$G_f(q^{-1})G_a(q^{-1}) = G(q^{-1}) \quad (30)$$

leading to the following conditions:

$$\sum_{k=1}^m g_{i,k}^f(q^{-1})g_{k,j}^a(q^{-1}) = g_{i,j}(q^{-1}) \quad i = 1, \dots, n \quad j = 1, \dots, m \quad (31)$$

that can be used to calculate the elements  $g_{k,j}^a(q^{-1})$  of the virtual actuator transfer matrix.

Notice that, for a SISO system, (31) reduces to the following condition:

$$g_a(q^{-1}) = g_f^{-1}(q^{-1})g(q^{-1}) \quad (32)$$

On the other hand, in the case of a fault appearing in the sensors changing their transfer function from  $H(q^{-1})$  to  $H_f(q^{-1})$ , the virtual sensor  $G_s(q^{-1})$  is placed after the faulty sensor (see Fig. 2) and is designed to satisfy:

$$G_s(q^{-1})H_f(q^{-1}) = H(q^{-1}) \quad (33)$$

leading to the conditions:

$$\sum_{k=1}^m g_{i,k}^s(q^{-1})h_{k,j}^f(q^{-1}) = h_{i,j}(q^{-1}) \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (34)$$

that can be used to calculate the elements  $g_{i,k}^s(q^{-1})$  of the virtual sensor transfer matrix.

Notice that looking at (30) and (33), the virtual actuator/sensor design problem can be assimilated to an *exact model matching problem* (Kaczorek, 1982). Necessary and sufficient conditions for the existence of solutions  $G_a(q^{-1})$  and  $G_s(q^{-1})$  are (Chen, 1984):

$$\text{rank } G_f(q^{-1}) = \text{rank} \begin{pmatrix} G_f(q^{-1}) & G(q^{-1}) \end{pmatrix} \quad (35)$$

and:

$$\text{rank } H_f(q^{-1}) = \text{rank} \begin{pmatrix} H_f(q^{-1}) \\ H(q^{-1}) \end{pmatrix} \quad (36)$$

over the field of rational functions of  $q^{-1}$  with coefficients in  $\mathbb{R}$ . A method for solving the *minimal design problem*, i.e. finding a proper transfer matrix with a minimal degree that solves the exact model matching problem, is described in Chen (1984).

In case of total failures of either an actuator or a sensor, conditions (30) and (33) could not be satisfied, because it could be impossible to recover the non-faulty transfer matrix using a virtual sensor and actuator approach. This is due to the fact that after the fault there is not enough hardware redundancy to compensate the faulty component with the remaining actuators/sensors. However, in cases where redundant actuators/sensors are available, the *hardware redundancy* can be exploited so as to achieve fault tolerance, by replacing the lost actuator/sensor

with the redundant ones. In this case, some additional action is required and it is useful to split the virtual actuator matrix in two blocks:

$$G_a(q^{-1}) = G_a^G(q^{-1})G_a^K(q^{-1}) \quad (37)$$

where  $G_a^G(q^{-1})$  is the part of the virtual actuator that has the function of trying to reproduce the non-faulty system behavior, while  $G_a^K(q^{-1})$  is the part of the virtual actuator that redistributes the lost actuator control action on the redundant ones.

For example, consider the following system with  $(m + 1)$  inputs and  $n$  outputs:

$$G(q^{-1}) = \begin{pmatrix} g_{1,1}(q^{-1}) & g_{1,1}(q^{-1}) & g_{1,2}(q^{-1}) & \dots & g_{1,m}(q^{-1}) \\ g_{2,1}(q^{-1}) & g_{2,1}(q^{-1}) & g_{2,2}(q^{-1}) & \dots & g_{2,m}(q^{-1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{n,1}(q^{-1}) & g_{n,1}(q^{-1}) & g_{n,2}(q^{-1}) & \dots & g_{n,m}(q^{-1}) \end{pmatrix} \quad (38)$$

where the first and the second input affect the outputs in the same way. Then, in case of faults, among which the total loss of an actuator, for example the first, the system transfer matrix changes to:

$$G_f(q^{-1}) = \begin{pmatrix} 0 & g_{1,1}^f(q^{-1}) & g_{1,2}^f(q^{-1}) & \dots & g_{1,m}^f(q^{-1}) \\ 0 & g_{2,1}^f(q^{-1}) & g_{2,2}^f(q^{-1}) & \dots & g_{2,m}^f(q^{-1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & g_{n,1}^f(q^{-1}) & g_{n,2}^f(q^{-1}) & \dots & g_{n,m}^f(q^{-1}) \end{pmatrix} \quad (39)$$

and the resulting virtual actuator transfer matrix is the following:

$$G_a(q^{-1}) = G_a^G(q^{-1})G_a^K(q^{-1}) = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ g_{1,1}^a(q^{-1}) & g_{1,1}^a(q^{-1}) & g_{1,2}^a(q^{-1}) & \dots & g_{1,m}^a(q^{-1}) \\ g_{2,1}^a(q^{-1}) & g_{2,1}^a(q^{-1}) & g_{2,2}^a(q^{-1}) & \dots & g_{2,m}^a(q^{-1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{m,1}^a(q^{-1}) & g_{m,1}^a(q^{-1}) & g_{m,2}^a(q^{-1}) & \dots & g_{m,m}^a(q^{-1}) \end{pmatrix} \quad (40)$$

while (30) is written as:

$$G_f(q^{-1})G_a^G(q^{-1}) = G_0(q^{-1}) = \begin{pmatrix} 0 & g_{1,1}(q^{-1}) & g_{1,2}(q^{-1}) & \dots & g_{1,m}(q^{-1}) \\ 0 & g_{2,1}(q^{-1}) & g_{2,2}(q^{-1}) & \dots & g_{2,m}(q^{-1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & g_{n,1}(q^{-1}) & g_{n,2}(q^{-1}) & \dots & g_{n,m}(q^{-1}) \end{pmatrix} \quad (41)$$



where  $G_0(q^{-1})$  is the matrix obtained from the nominal system transfer matrix by considering the effect of the actuator total loss. A similar approach is used in the case of sensor loss.

**Remark 3.1.** There exists a connection between the rank conditions (35)-(36) and their counterpart in the case of virtual actuators and sensors designed in state space form following the approach proposed by Lunze & Steffen (2003). In fact, in this latter case, if the fault appearance has changed the state space matrices from  $(A, B, C)$  to  $(A, B_f, C)$  in the case of actuator faults, or to  $(A, B, C_f)$  in the case of sensor faults, it is possible to recover the nominal system behavior using a static compensation if and only if (Blanke et al., 2006):

$$\text{rank } B_f = \text{rank} \begin{pmatrix} B_f & B \end{pmatrix} \quad (42)$$

or:

$$\text{rank } C_f = \text{rank} \begin{pmatrix} C_f \\ C \end{pmatrix} \quad (43)$$

respectively. Then, if we consider the following realizations of  $G(q^{-1})$  and  $H(q^{-1})$ :

$$G(q^{-1}) = (q^{-1}I - A_G)^{-1} B \quad (44)$$

$$H(q^{-1}) = C (q^{-1}I - A_H)^{-1} \quad (45)$$

and only the faults affecting the matrices  $B$  and  $C$ , as follows:

$$G_f(q^{-1}) = (q^{-1}I - A_G)^{-1} B_f \quad (46)$$

$$H_f(q^{-1}) = C_f (q^{-1}I - A_H)^{-1} \quad (47)$$

the rank conditions (35)-(36) are equivalent to (42)-(43). In cases where (42) or (43) were not satisfied, fault tolerance could still be achieved, even though at the price of using a dynamical virtual actuator or virtual sensor, thus introducing additional poles in the system.

## 4. Case study: Wind Turbine System

### 4.1. Wind Turbine Model

The wind turbine model of the FDI/FTC benchmark described in Odgaard et al. (2013) comprises the Wind model, the Blade and Pitch model, the Drive Train model, the Generator/Converter model and the Controller.

The hydraulic pitch system can be modeled as (Merritt, 1967):

$$\frac{\beta(s)}{\beta_r(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (48)$$

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

The generator/converter dynamics can be modeled by:

$$\frac{\tau_g(s)}{\tau_{g,r}(s)} = \frac{\alpha_{gc}}{s + \alpha_{gc}} \quad (49)$$

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

The power produced by the generator  $P_g$  depends on the rotational speed of the generator  $\omega_g$  and the applied load  $\tau_g$ :

$$P_g(t) = \eta_g \omega_g(t) \tau_g(t) \quad (50)$$

where  $\eta_g$  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  $B_g$ . The shafts are interconnected by a transmission having a gear ratio  $N_g$  and an efficiency  $\eta_{dt}$ , combined with a torsion stiffness  $K_{dt}$ , and a torsion damping  $B_{dt}$ . The model is described by the following three differential equations (Sloth et al., 2011):

$$\dot{\omega}_r(t) = -\frac{(B_{dt} + B_r)}{J_r} \omega_r(t) + \frac{B_{dt}}{N_g J_r} \omega_g(t) - \frac{K_{dt}}{J_r} \theta_{\Delta}(t) + \frac{\tau_r(t)}{J_r} \quad (51)$$

$$\dot{\omega}_g(t) = \frac{\eta_{dt} B_{dt}}{N_g J_g} \omega_r(t) - \left( \frac{\eta_{dt} B_{dt}}{N_g^2 J_g} + \frac{B_g}{J_g} \right) \omega_g(t) + \frac{\eta_{dt} K_{dt}}{N_g J_g} \theta_{\Delta}(t) - \frac{\tau_g(t)}{J_g} \quad (52)$$

$$\dot{\theta}_{\Delta}(t) = \omega_r(t) - \frac{\omega_g(t)}{N_g} \quad (53)$$

where  $\omega_r$  is the rotor speed,  $\omega_g$  is the generator speed,  $\theta_{\Delta}$  is the torsion angle of the drive train,  $\tau_r$  is the aerodynamic torque and  $\tau_g$  is the generator torque.

Since the turbine has three blades, all three pitch positions are measured. Such measurements are done with two sensors in order to ensure physical redundancy (variables defined as  $\beta_{r1}, \beta_{r2}, \beta_{r3}$  for the pitch reference to blade 1, 2 and 3;  $\beta_{1,m1}, \beta_{1,m2}, \beta_{2,m1}, \beta_{2,m2}, \beta_{3,m1}, \beta_{3,m2}$  are the pitch positions measurements starting from

Table 1: Wind Turbine Fault Description

Fault	Fault description	Type	Value	Period
$f_1$	Change in pitch 1 measurement	Fixed Value	$\beta_{1,m1} = 5^\circ$	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	$\beta_{3,m1} = 10^\circ$	2600s-2700s
$f_4$	Change in Rotor speed sensor	Fixed value	$\omega_{r,m1} = 1.4rad/s$	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

the two measurements for the blade 1 followed by the two measurements for blade 2 and blade 3 in the end). The generator and rotor speeds are also measured with two sensors each for the same reason (the two rotor speed measurements are defined as  $\omega_{r,m1}$ ,  $\omega_{r,m2}$ , while the two generator speed measurements are defined as  $\omega_{g,m1}$ ,  $\omega_{g,m2}$ ). Details about the controller can be found in Odgaard et al. (2013).

In this paper, the different faults proposed in the FDI/FTC benchmark (Odgaard et al., 2013) will be considered, as resumed in Table 1.

#### 4.2. Residual Generation

According to Blesa et al. (2011), after applying structural analysis (Blanke et al., 2006) with the aid of the SaTool (Blanke & Lorentz, 2006) to the set of equations provided in Odgaard et al. (2013), the following set of twelve residuals expressed as reduced observers in input-output form can be obtained:

$$\begin{aligned}
r_1(k) &= \omega_{r,m1}(k) - \omega_{r,m2}(k) \\
r_2(k) &= \left(1 - \frac{l_{2,1}q^{-1}}{1 - (a_{2,1} - l_{2,1})q^{-1}}\right)\omega_{r,m2}(k) - \frac{b_{2,1}q^{-1}\tau_r(k) + c_{2,1}q^{-1}\tau_{g,m}(k)}{1 - (a_{2,1} - l_{2,1})q^{-1}} \\
r_3(k) &= \omega_{g,m1}(k) - \omega_{g,m2}(k) \\
r_4(k) &= \left(1 - \frac{l_{4,1}q^{-1}}{1 - (a_{4,1} - l_{4,1})q^{-1}}\right)\omega_{g,m2}(k) - \frac{b_{4,1}q^{-1}\tau_r(k) + c_{4,1}q^{-1}\tau_{g,m}(k)}{1 - (a_{4,1} - l_{4,1})q^{-1}} \\
r_5(k) &= \beta_{1,m1}(k) - \beta_{1,m2}(k) \\
r_6(k) &= \left(1 - \frac{l_{6,1}q^{-1} + l_{6,2}q^{-2}}{1 - (a_{6,1} - l_{6,1})q^{-1} - (a_{6,2} - l_{6,2})q^{-2}}\right)\beta_{1,m2}(k) \\
&\quad - \frac{(b_{6,1}q^{-1} + b_{6,2}q^{-2})\beta_r(k)}{1 - (a_{6,1} - l_{6,1})q^{-1} - (a_{6,2} - l_{6,2})q^{-2}} \\
r_7(k) &= \beta_{2,m1}(k) - \beta_{2,m2}(k) \\
r_8(k) &= \left(1 - \frac{l_{8,1}q^{-1} + l_{8,2}q^{-2}}{1 - (a_{8,1} - l_{8,1})q^{-1} - (a_{8,2} - l_{8,2})q^{-2}}\right)\beta_{2,m2}(k) \\
&\quad - \frac{(b_{8,1}q^{-1} + b_{8,2}q^{-2})\beta_r(k)}{1 - (a_{8,1} - l_{8,1})q^{-1} - (a_{8,2} - l_{8,2})q^{-2}}
\end{aligned} \tag{54}$$

$$\begin{aligned}
r_9(k) &= \beta_{3,m1}(k) - \beta_{3,m2}(k) \\
r_{10}(k) &= \left(1 - \frac{l_{10,1}q^{-1} + l_{10,2}q^{-2}}{1 - (a_{10,1} - l_{10,1})q^{-1} - (a_{10,2} - l_{10,2})q^{-2}}\right)\beta_{3,m2}(k) \\
&\quad - \frac{(b_{10,1}q^{-1} + b_{10,2}q^{-2})\beta_r(k)}{1 - (a_{10,1} - l_{10,1})q^{-1} - (a_{10,2} - l_{10,2})q^{-2}} \\
r_{11}(k) &= \left(1 - \frac{l_{11,1}q^{-1}}{1 - (a_{11,1} - l_{11,1})q^{-1}}\right)\tau_{g,m}(k) - \frac{b_{11,1}q^{-1}\tau_{g,r}(k)}{1 - (a_{11,1} - l_{11,1})q^{-1}} \\
r_{12}(k) &= P_{g,m}(k) - \eta_g\omega_{g,m2}(k)\tau_{g,m}(k)
\end{aligned}$$

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

$$\theta = \left( \begin{array}{cccccccccccc}
a_{2,1} & b_{2,1} & c_{2,1} & a_{4,1} & b_{4,1} & c_{4,1} & a_{6,1} & a_{6,2} & b_{6,1} & b_{6,2} \\
a_{8,1} & a_{8,2} & b_{8,1} & b_{8,2} & a_{10,1} & a_{10,2} & b_{10,1} & b_{10,2} & a_{11,1} & b_{11,1}
\end{array} \right)^T \tag{55}$$

Table 2: 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			

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 non-linearity is hidden in some of the residuals due to the use of the variable  $\tau_r$ , which is estimated using the following relation given in Odgaard et al. (2013):

$$\tau_r(k) = \rho\pi R^3 C_q(\lambda(k), \beta(k)) v_w(k)^2 / 2. \quad (56)$$

Moreover, SaTool provides the FSM represented in Table 2, 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 2.5,  $m_{i,j} = 1$  where there is a cross,  $m_{i,j} = 0$  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 Svärd & Nyberg (2012).

#### 4.3. Uncertain parameter estimation

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, 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, residual  $r_{11}$  is derived from the converter dynamics model.

In order to reduce the effect of the noise in fault detection and isolation procedures, the variables involved in the discretized regressor equations have been filtered by second order low-pass filters. Nominal coefficients of dynamical residuals have been obtained applying the least squares method to the data obtained simulating the system in the fault free scenario. Once the nominal parameters have been estimated, the observer gains  $l_{i,j}$  in the dynamical residuals are determined taking into account that there is a trade-off between the observer convergence towards the real system behavior and the fault indication persistence (Meseguer et al., 2010). Changing the observer gain, observer poles can be shifted from a location close to the ones of the system to a location close to the origin of the  $z$ -plane. In particular, selecting the observer poles close to the ones of the system, the persistence of the fault indication increases although the convergence towards the real system behavior becomes slower. This is the location that has been used in this paper. The use of this criterion has been motivated by the results that were previously obtained in Blesa et al. (2011), where a deadbeat observer (predictor) was used, that were not completely satisfactory (some of the faults in the wind turbine benchmark were not detected, e.g.  $f_6$ , while others were detected with a big delay, e.g.  $f_7$ ). Notice that the deadbeat observer corresponds to the fastest possible observer since it places the poles in the origin of the  $z$ -plane.

The uncertain parameter estimation procedure described in Section 2.3 has been applied to the fault-free scenario specified in the benchmark in order to obtain the intervals for the parameters of the residuals that will be used for fault detection and isolation purposes. The noise bound  $\sigma$  is estimated using three times the standard deviation of the nominal residual (21). On the other hand, the parameter uncertainty bounds are estimated using the procedure described in Section 2.3. This procedure guarantees that all data collected in non-faulty scenarios used for estimation will be contained in the interval output produced by the interval observer, avoiding in this way false alarms. After estimating the uncertain parameters, the stability and the pole location of the observer are checked using the LMIs of the regional pole placement approach proposed by Chilali & Gahinet (1996), with a circular  $\mathcal{D}$  region centered at the origin of the  $z$ -plane and with radius long enough to contain the poles of the system. In this way, it is guaranteed that the observer will not be slower than the plant.

#### 4.4. Fault Detection and Isolation

Fault detection has been implemented by means of (23) applied to the 12 residuals obtained in Section 4.2. Fault isolation has been implemented using the Algorithm 2 with the FSM presented in Table 2.

In order to improve the fault detection and isolation results the following practical enhancements could be introduced to the previous schemes:

- Pre-processing sensor data in order to detect stuck faults based on comparing the current and previous measurements. If the same value is observed, then the residuals that use these measurements are considered as violated.
- After a gain sensor fault is detected and isolated, if the sensor takes measurements close to zero (within the interval  $[-10\sigma, 10\sigma]$ ), the residual is kept active. This trick is done to avoid that the fault is not persistently detected and isolated.
- After a parametric fault in a dynamical residual is detected and isolated, the fault detection test (23) is complemented with a test based on estimating the residual parameters using the algorithm described in Section 2.6. This complementary test checks if the residual parameters are inside a confidence interval around the nominal parameter values. As in the previous case, this enhancement helps the persistent fault detection and isolation.

#### 4.5. Fault Estimation

When a fault affects a sensor, this could be of two different natures depending on if the sensor is stuck at a fixed value or is affected by a change of the gain. In both cases, fault isolation could be enough to achieve fault tolerance by deactivating the faulty sensor and using the redundant healthy one. However, in the change of gain case (multiplicative fault) the information coming from the faulty sensor could be recovered if an estimation of the fault is available, improving the performance of the control system. In this case, the regression equation is obtained from the measurements of the redundant sensors, namely  $y_{m,1}(k)$  and  $y_{m,2}(k)$ . For example, in case of a fault in the first sensor, (26) is given by:

$$y_{m,1}(k) = y_{m,2}(k)\kappa \quad (57)$$

where  $\kappa$  is the multiplicative sensor fault gain factor.

Faults in the hydraulic pitch system change the values of  $\omega_n$  and  $\zeta$  in (48), that can be rewritten in the following differential form:

$$\ddot{\beta}(t) + 2\zeta\omega_n\dot{\beta}(t) + \omega_n^2\beta(t) = \omega_n^2\beta_r(t) \quad (58)$$

which can be discretized using an Euler approximation and then brought to the following regression form to be used for fault estimation:

$$\ddot{\beta}(k-2) = \begin{bmatrix} \omega_n^2 & \omega_n\zeta \end{bmatrix} \begin{bmatrix} \beta_r(k-2) - \beta(k-2) \\ -2\dot{\beta}(k-2) \end{bmatrix} \quad (59)$$

with:

$$\dot{\beta}(k-2) \cong \frac{\beta(k-1) - \beta(k-2)}{T_s} \quad (60)$$

$$\ddot{\beta}(k-2) \cong \frac{\beta(k) - 2\beta(k-1) + \beta(k-2)}{T_s} \quad (61)$$

Finally, the fault in the converter consists in an offset that can be estimated using the following linear regression:

$$\tau_{g,m}(k) - \hat{\tau}_g(k) = \Delta\tau_g \quad (62)$$

where  $\Delta\tau_g$  is the offset and  $\hat{\tau}_g$  is the expected converter torque obtained from its model (49).

#### 4.6. Fault Tolerant Control

The idea behind the virtual actuators and sensors approach is that one can achieve fault tolerance without touching the nominal controller that was designed to work in non-faulty conditions. The nominal controller has the following inputs:  $P_g$ ,  $P_r$  and  $\omega_g$ , and the following outputs:  $\beta_{r1}$ ,  $\beta_{r2}$ ,  $\beta_{r3}$  and  $\tau_{g,r}$ .

*Fault 1* - The fault 1 is a stuck value,  $\beta_{1,m1} = 5^\circ$ , in the time period 2000s-2100s. The input to the first hydraulic pitch system is given by the following expression:

$$u_1 = \beta_{1,r} + \beta_{1,f} \quad (63)$$

where  $\beta_{1,r}$  is the controller output and  $\beta_{1,f} = \beta_1 - 0.5(\beta_{1,m1} + \beta_{1,m2})$  with  $\beta_1$  the real angle of the first pitch system, and  $\beta_{1,m1}$ ,  $\beta_{1,m2}$  the values given by the sensors. In the case of sensor failure to a stuck value, it is desired to eliminate the faulty sensor from the overall control loop. In this case, the fault tolerance is achieved



by adding an extra term to the input (63) such that its effect equals the fault effect, that is:

$$u_1^{FTC} = \beta_{1,r} + \Delta\beta_{1,r} + \beta_1 - 0.5\beta_{1,m1} - 0.5\beta_{1,m2} = \beta_{1,r} + \beta_1 - \beta_{1,m2} \quad (64)$$

Eq. (64) leads to the result that:

$$\Delta\beta_{1,r} = 0.5(\beta_{1,m1} - \beta_{1,m2}) \quad (65)$$

*Fault 2* - The fault 2 is a change of gain  $\beta_{2,m2} = \kappa_2\beta_{2,m2}$  with  $\kappa_2 = 1.2$ , in the time period 2300s-2400s. In this case, the input to the second hydraulic pitch system is given by an expression analogous to (63):

$$u_2 = \beta_{2,r} + \beta_{2,f} \quad (66)$$

The fault tolerance is achieved by adding an extra term to the input (66), in order to eliminate the effect of the faulty sensor change of gain on the overall control loop, such that:

$$u_2^{FTC} = \beta_{2,r} + \Delta\beta_{2,r} + \beta_2 - 0.5(\beta_{2,m1} + \kappa_2\beta_{2,m2}) = \beta_{2,r} + \beta_2 - 0.5(\beta_{2,m1} + \beta_{2,m2}) \quad (67)$$

that leads to:

$$\Delta\beta_{2,r} = 0.5(\hat{\kappa}_2 - 1)\beta_{2,m2} \quad (68)$$

where  $\hat{\kappa}_2$  is the value obtained through the fault estimation method.

*Fault 3* - The fault 3 is a stuck value,  $\beta_{3,m1} = 10^\circ$ , in the time period 2600s-2700s. This fault is very similar to the first one, and the fault tolerance is obtained in a similar way by adding the term:

$$\Delta\beta_{3,r} = 0.5(\beta_{3,m1} - \beta_{3,m2}) \quad (69)$$

*Fault 4* - The fault 4 is a stuck value,  $\omega_{r,m1} = 1.4rad/s$ , in the time period 1500s-1600s. As the measurements  $\omega_{r,m1}$  and  $\omega_{r,m2}$  are not used by the nominal controller, no fault tolerance action is needed.

*Fault 5* - The fault 5 is a change of gains of two sensors:  $\omega_{r,m2} = \kappa_r\omega_{r,m2}$  and  $\omega_{g,m1} = \kappa_g\omega_{g,m1}$ , with  $\kappa_r = 1.1$  and  $\kappa_g = 0.9$ , in time period 1000s-1100s. No action on the rotor speed measurement is needed, while fault tolerance for the change of gain in the generator speed is achieved in a very similar way to fault 2.

In this case, the nominal controller uses a value  $\omega_{g,m}$  that is given by:

$$\omega_{g,m} = \omega_{g,m1} + \omega_{g,m2} \quad (70)$$

An additive term can be added so as to compensate the fault effect, that is:

$$0.5(\omega_{g,m1} + \kappa_g \omega_{g,m2}) + \Delta\omega_{g,m} = 0.5(\omega_{g,m1} + \omega_{g,m2}) \quad (71)$$

resulting in the expression for the additive term:

$$\Delta\omega_{g,m} = 0.5(1 - \hat{\kappa}_g)\omega_{g,m2} \quad (72)$$

where  $\hat{\kappa}_g$  is the value obtained through the fault estimation method.

*Fault 6* - This fault is an abrupt change of the parameters that describe the pitch actuator 2 dynamics,  $\omega_{n2}$  and  $\zeta_2$ . Fault tolerance is obtained as follows:

$$\beta_{r,2}^{FTC}(s) = \frac{s^2 + 2\hat{\zeta}_2\hat{\omega}_{n2}s + \hat{\omega}_{n2}^2}{s^2 + 2\zeta_2\omega_{n2}s + \omega_{n2}^2}\beta_{r,2}(s) \quad (73)$$

Since the fault tolerant control system works in discrete-time, a sample and hold discretization is performed on  $\beta_{r,2}^{FTC}(s)$  in order to obtain  $\beta_{r,2}^{FTC}(z)$ .

*Fault 7* - This fault is an incipient change of the parameters that describe the pitch actuator 3 dynamics,  $\omega_{n3}$  and  $\zeta_3$ . Fault tolerance is obtained as follows:

$$\beta_{r,3}^{FTC}(s) = \frac{s^2 + 2\hat{\zeta}_3\hat{\omega}_{n3}s + \hat{\omega}_{n3}^2}{s^2 + 2\zeta_3\omega_{n3}s + \omega_{n3}^2}\beta_{r,3}(s) \quad (74)$$

Since the fault tolerant control system works in discrete-time, a sample and hold discretization is performed on  $\beta_{r,3}^{FTC}(s)$  in order to obtain  $\beta_{r,3}^{FTC}(z)$ .

*Fault 8* - The fault 8 is an offset in the converter system  $\tau_g = \tau_g + \Delta\tau_g$  with  $\tau_g = 2000Nm$  from 3800s to 3900s. In this case, the fault tolerance is achieved by subtracting the estimated value  $\Delta\hat{\tau}_g$  from the converter system reference  $\tau_{g,r}$ :

$$\tau_{g,r}^{FTC} = \tau_{g,r} - \Delta\hat{\tau}_{g,r} \quad (75)$$

## 5. Results

The proposed FDI strategy has been applied to the Benchmark proposed in Odgaard et al. (2013). The parameter bounds that determine the box  $\Theta$  and the observer gains that have been obtained in the identification and design processes are presented in Table 3.

All the faults have been detected and isolated and it can be seen from the obtained results (see Table 4) that the FDI system satisfies almost all detection time requirements (Odgaard et al., 2013). On the other hand, the results achieved in

Residual	Model Parameters	Observer Gains
$r_2(k)$	$a_{2,1} \in [0.97699, 1.0067]$	$l_{2,1} = 0.49594$
	$b_{2,1} \in [8.54231, 8.80249] 10^{-10}$	
	$c_{2,1} \in [3.6891, 3.8014] 10^{-7}$	
$r_4(k)$	$a_{4,1} \in [0.98792, 1.0119]$	$l_{4,1} = 0.49996$
	$b_{4,1} \in [1.6357, 1.6755] 10^{-8}$	
	$c_{4,1} \in [-1.6240, -1.5855] 10^{-6}$	
	$a_{6,1} \in [1.8608, 1.8664]$	
$r_6(k)$	$a_{6,2} \in [-8.7650, -8.7387] 10^{-1}$	$l_{6,1} = 1.8636 \times 10^{-3}$
	$b_{6,1} \in [5.8913, 5.9091] 10^{-3}$	$l_{6,2} = -8.7518 \times 10^{-4}$
	$b_{6,2} \in [5.6352, 5.6522] 10^{-3}$	
	$a_{8,1} \in [1.8608, 1.8664]$	
$r_8(k)$	$a_{8,2} \in [-8.7650, -8.7387] 10^{-1}$	$l_{8,1} = 1.8636 \times 10^{-3}$
	$b_{8,1} \in [5.8913, 5.9091] 10^{-3}$	$l_{8,2} = -8.7518 \times 10^{-4}$
	$b_{8,2} \in [5.6352, 5.6522] 10^{-3}$	
	$a_{10,1} \in [1.8608, 1.8664]$	
$r_{10}(k)$	$a_{10,2} \in [-8.7650, -8.7387] 10^{-1}$	$l_{10,1} = 1.8636 \times 10^{-3}$
	$b_{10,1} \in [5.8913, 5.9091] 10^{-3}$	$l_{10,2} = -8.7518 \times 10^{-4}$
	$b_{10,2} \in [5.6352, 5.6522] 10^{-3}$	
$r_{11}(k)$	$a_{11,1} \in [0.60639, 0.60653]$	$l_{11,1} = 0.12131$
	$b_{11,1} \in [0.39338, 0.39356]$	

this work are equal or better than the ones obtained in the FDI Benchmark competition, which results are reported and compared in Odgaard et al. (2013). In fact, the method proposed in this paper requires equal or less detection time than some other approaches presented in Odgaard et al. (2013). The results obtained have also been compared to the ones presented in Casau et al. (2012), where an approach based on set-valued observers for FDI and FTC is used. The latter has shown a good performance in detecting almost all the faults proposed in the benchmark (Odgaard et al., 2013) except  $f_4$  and  $f_6$  that were detected later (5.61s and 4.615s, respectively). However, the time detection obtained in this work has been 0.03s and 0.06s for  $f_4$  and  $f_6$ , respectively.

#### *Fault Scenario 1*

Fig. 3 and Fig. 4 show the nominal residuals  $r_1^0, \dots, r_{12}^0$  and the observed fault signature components  $\phi_1, \dots, \phi_{12}$  used in the proposed fault detection and isolation procedure in fault scenario 1 (fixed value on pitch 1 position sensor from  $t = 2000s$  to  $t = 2100s$ ). The fault is detected at instant  $t = 2000.03s$ , as shown in

Table 4: FDI results for the set of considered fault scenarios.

Fault scenario	Fault appearance time	FDI time
1	2000s	2000.03s
2	2300s	2300.06s
3	2600s	2600.03s
4	1500s	1500.03s
5	1000s	1000.03s
6	2900s	2900.06s
7	3400s	3411.60s
8	3800s	3800.04s

Fig. 5.

Fig. 6 shows that the proposed FTC strategy results in an improvement of the generated power. Similar results have been obtained in fault scenario 4. In fault scenario 3, it has been noticed that the fault is not so critical as in the other cases, since the working point of the third pitch position nearly corresponds to the stuck value of the faulty sensor ( $10^\circ$ ). Thus, in fault scenario 3 no visible differences have been observed between the behavior with FTC and the one without FTC.

#### *Fault Scenario 2*

Fig. 7 and Fig. 8 show the nominal residuals  $r_1^0, \dots, r_{12}^0$  and the observed fault signature components  $\phi_1, \dots, \phi_{12}$  used in the proposed fault detection and isolation procedure in fault scenario 2 (change of gain in the second pitch measurement from  $t = 2300s$  to  $t = 2400s$ ). The fault is detected at instant  $t = 2300.06s$ , as shown in Fig. 9.

If the fault results in a change of sensor gain, as in the case of faults 2 and 5, the tolerance is obtained multiplying the faulty output by the inverse of the gain factor. Thus, the change in gain due to the fault must be estimated and the performance of the proposed FTC strategy depends directly on such estimation. The better the estimation is, the lower is the loss of performance due to the fault occurrence. Hereafter, results obtained in the second fault scenario are shown. Fig. 10 demonstrates the results with respect to the estimation, obtained by applying the proposed least-square approach with  $N = 10$ . This value has been found as a good trade-off between a lower value that would have made faster the estimation at the expense of higher sensitivity to noise and a higher value that would have made the estimation more robust to the noise but also slower. It is worth remarking that a low-pass filter has been applied to the sensor data in order to remove some noise

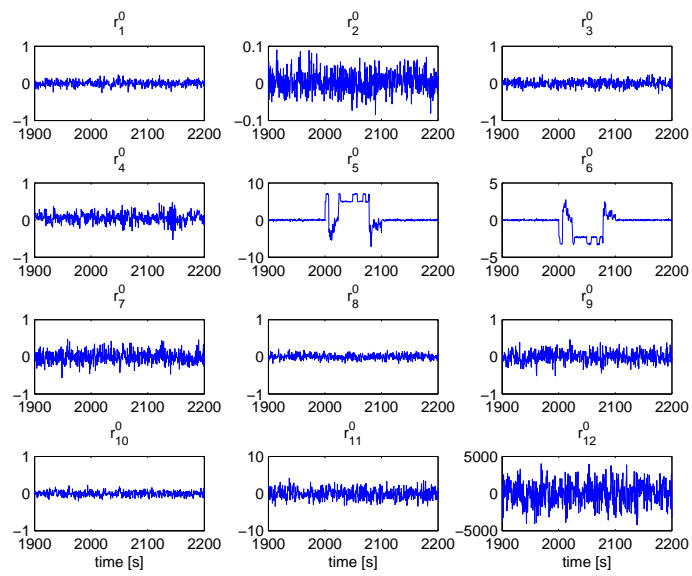


Figure 3: Nominal residuals  $r_1^0, \dots, r_{12}^0$  in *fault scenario 1*.

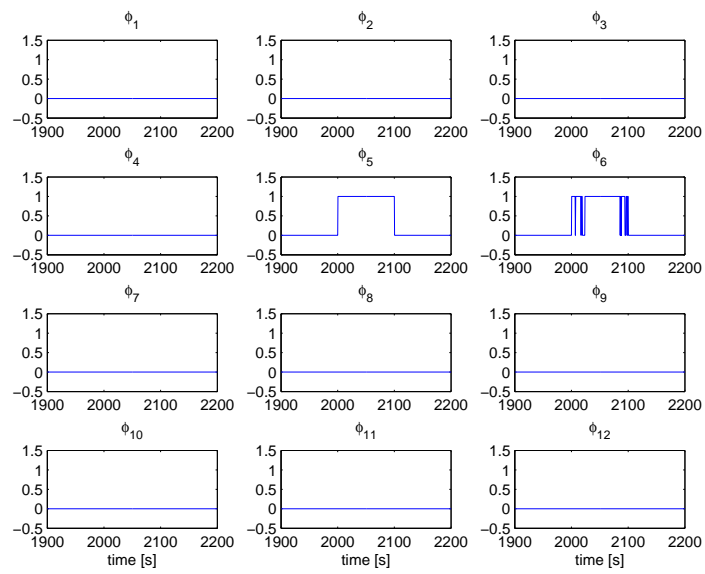


Figure 4: Observed fault signature components in *fault scenario 1*.

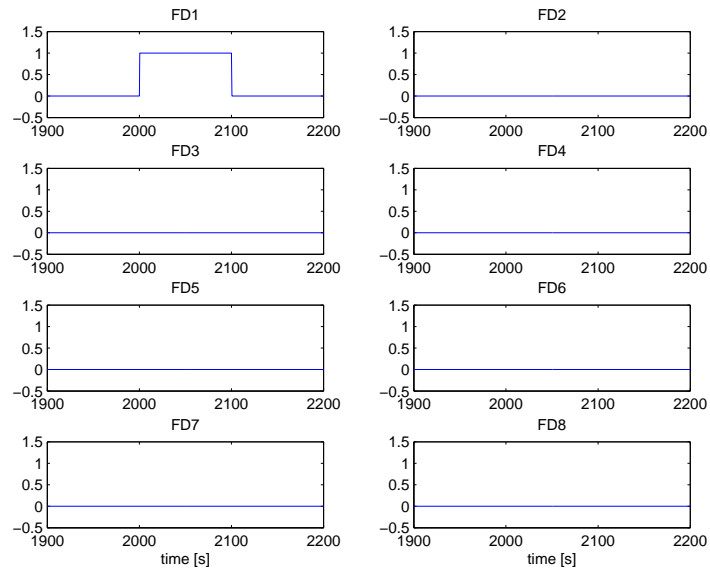


Figure 5: Fault detection results in *fault scenario 1*.

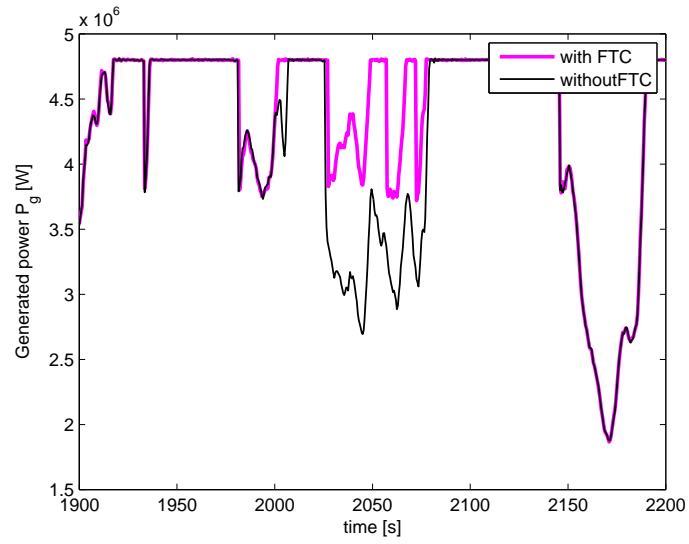


Figure 6: Generated power  $P_g$  with and without FTC in *fault scenario 1*.

before applying the fault estimation algorithm. Fig. 11 shows the effectiveness of the proposed strategy. In this case, the performances with FTC are slightly better than the ones obtained without FTC. Similar results have been obtained in fault scenario 5.

#### *Fault Scenario 6*

Fig. 12 and Fig. 13 show the nominal residuals  $r_1^0, \dots, r_{12}^0$  and the observed fault signature components  $\phi_1, \dots, \phi_{12}$  used in the proposed fault detection and isolation procedure in fault scenario 6 (abrupt change in the second order transfer function parameters  $\omega_n$  and  $\zeta$ ). The fault is detected at instant  $t = 2900.06s$ , as shown in Fig. 14.

In fault scenarios 6 and 7, the fault causes a change in the second order transfer function parameters  $\omega_n$  and  $\zeta$ . Such a change is abrupt in the sixth fault scenario and incipient in the seventh scenario. The estimation of these parameters in fault scenario 6 is shown in Fig. 15 and Fig. 16, respectively. It can be seen that a good estimation of the natural frequency is achieved, while the algorithm does not provide an accurate estimation of the damping ratio. The reason is probably due to the effect of the low-pass filter used to remove noise from data before applying the fault estimation algorithm.

However, in spite of the difficulty in estimating the real value of  $\zeta_2$ , the quality of the fault estimation is good enough to allow the active fault tolerance mechanism based on a virtual actuator to compensate the fault, as shown in Fig. 17. In this figure, such a behavior corresponds to the pitch response getting as near as possible to the nominal response (blue line). It can be seen that the response when no fault tolerance mechanism acts on the system (green line) is quite different with respect to the nominal one. The proposed FTC strategy can compensate the fault resulting in an almost perfect matching between the response with FTC (red line) and the nominal one.

#### *Fault Scenario 8*

Fig. 18 and Fig. 19 show the nominal residuals  $r_1^0, \dots, r_{12}^0$  and the observed fault signature components  $\phi_1, \dots, \phi_{12}$  used in the proposed fault detection and isolation procedure in fault scenario 8 (converter offset from  $t = 3800s$  to  $t = 3900s$ ). The fault is detected at instant  $t = 3800.04s$ , as shown in Fig. 20.

In Fig. 21, it can be seen that the generated power with FTC is approximately the same as its reference value (4.8 MW), and the offset brought in by the fault occurrence is corrected.



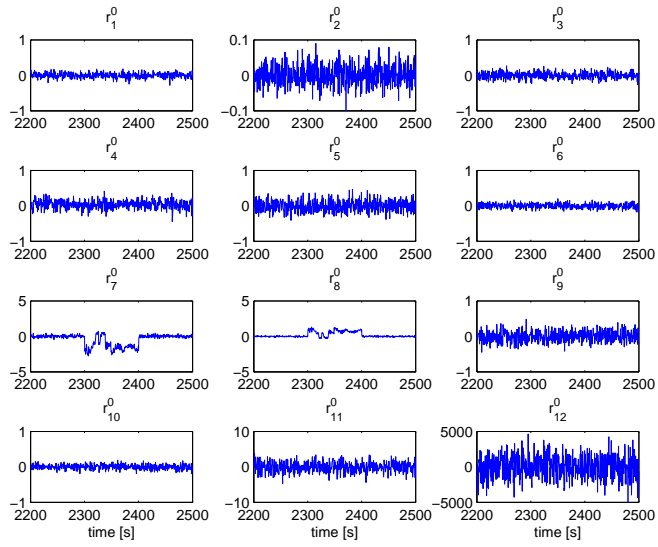


Figure 7: Nominal residuals  $r_1^0, \dots, r_{12}^0$  in *fault scenario 2*.

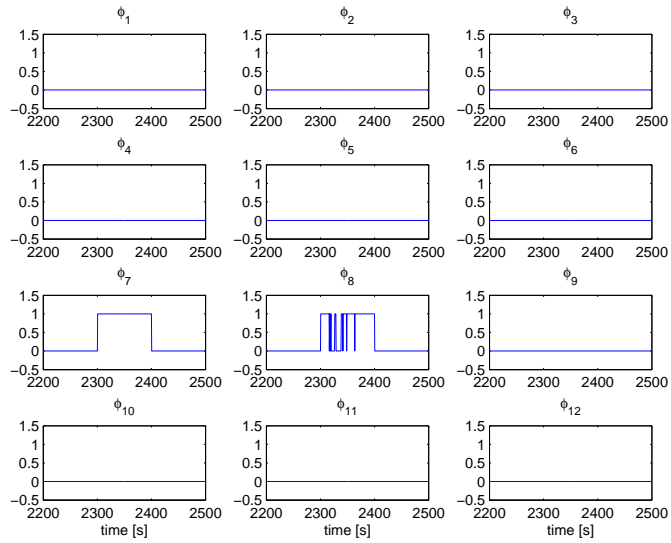


Figure 8: Observed fault signature components in *fault scenario 2*.

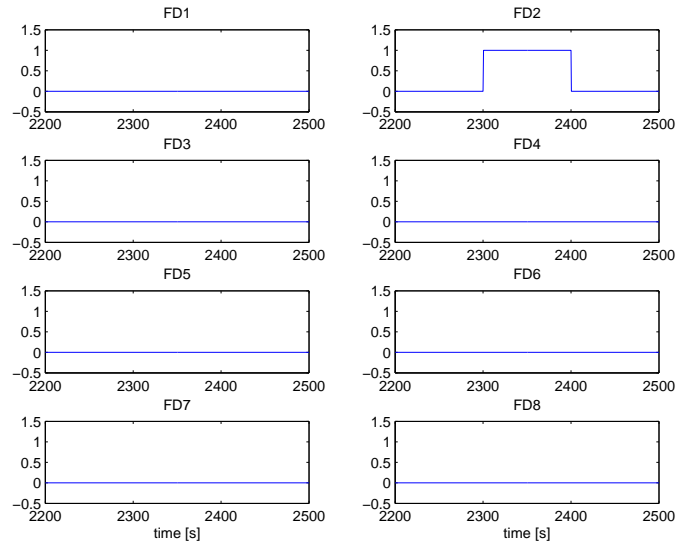


Figure 9: Fault detection results in *fault scenario 2*.

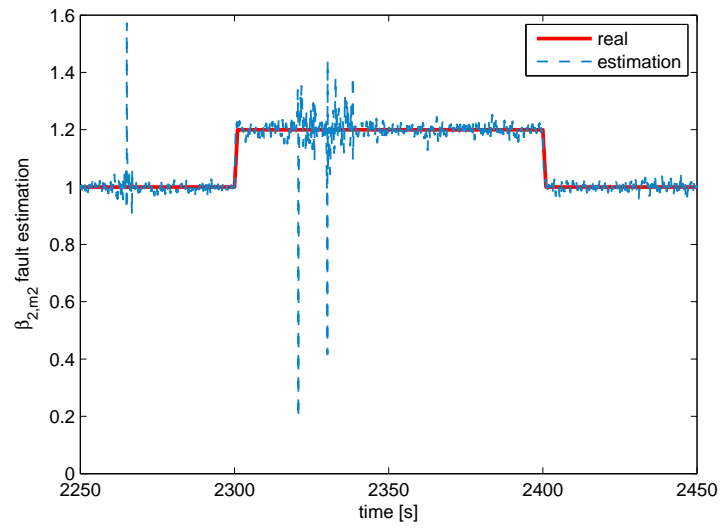


Figure 10: Change of gain estimation in *fault scenario 2*.

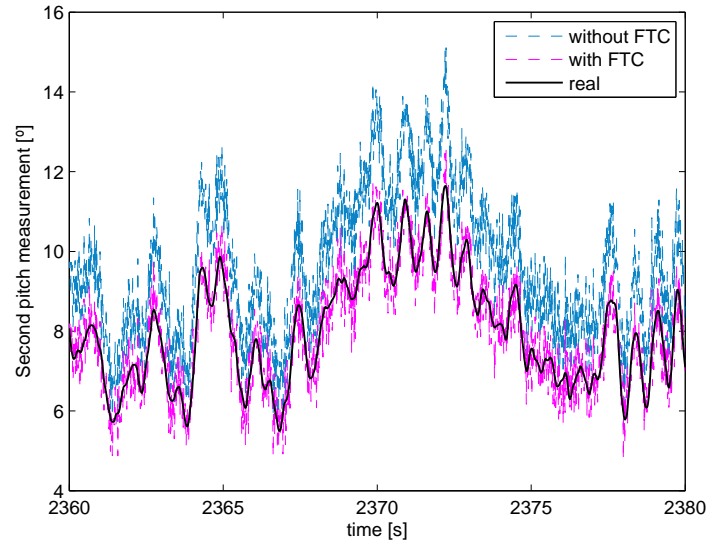


Figure 11: Measurement  $\beta_{2,m2}$  with and without FTC in *fault scenario 2*.

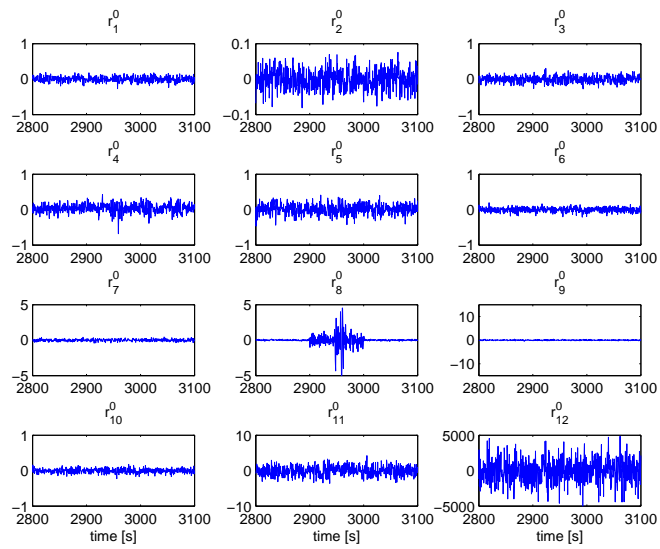


Figure 12: Nominal residuals  $r_1^0, \dots, r_{12}^0$  in *fault scenario 6*.

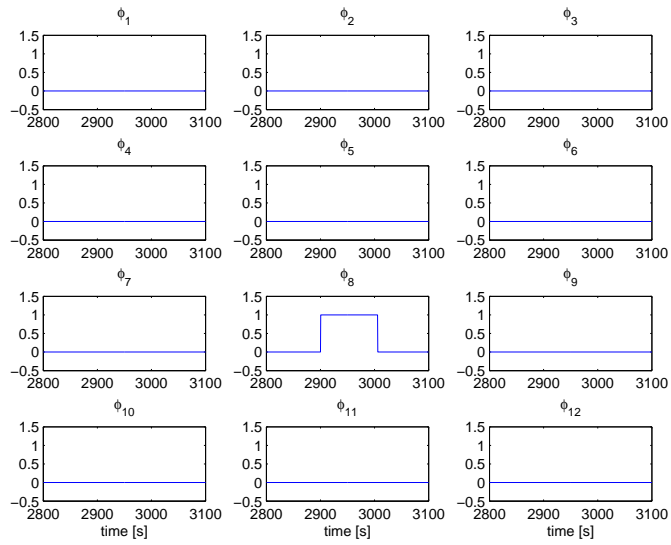


Figure 13: Observer fault signature components in *fault scenario 6*.

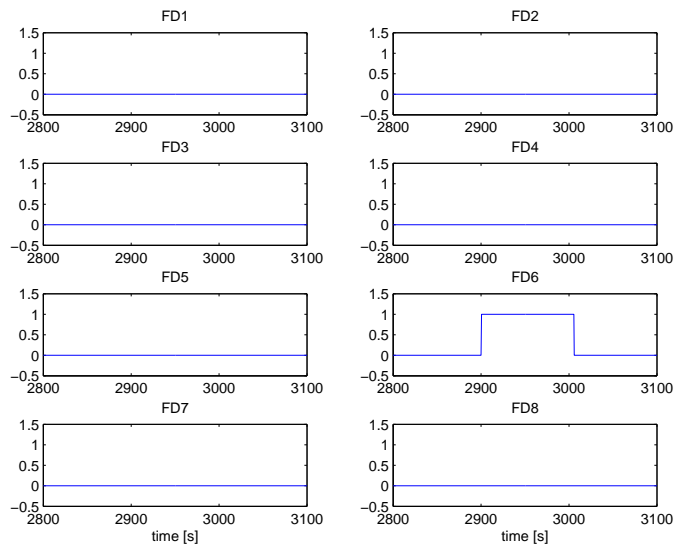


Figure 14: Fault detection results in *fault scenario 6*.

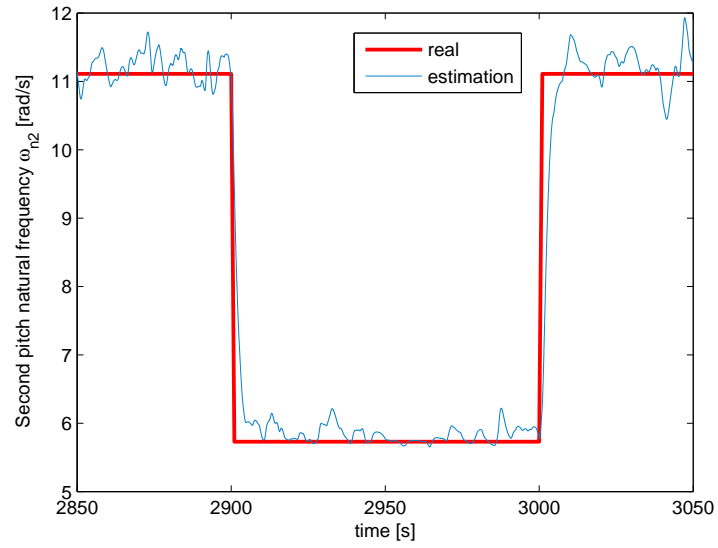


Figure 15:  $\omega_{n,2}$  estimation in *fault scenario 6*.

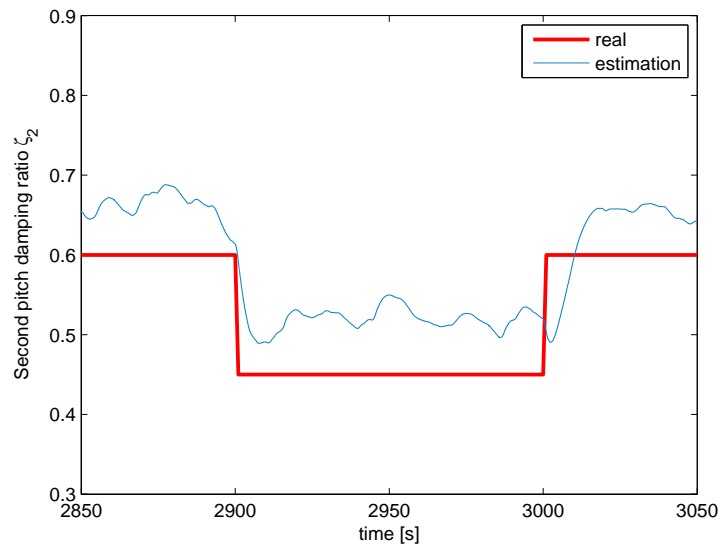


Figure 16:  $\zeta_2$  estimation in *fault scenario 6*.

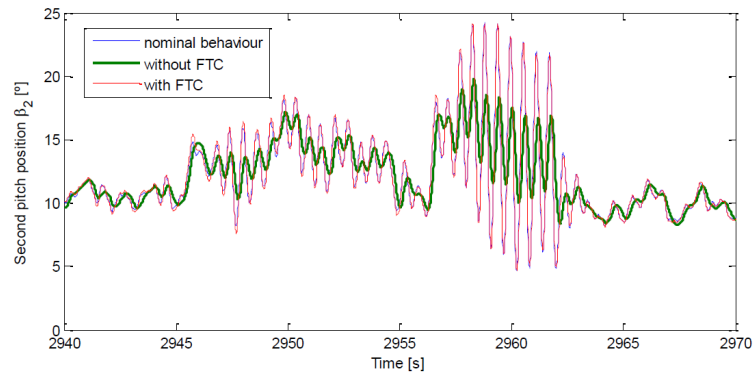


Figure 17:  $\beta_2$  response without and with FTC in *fault scenario 6*.

## 6. Conclusions

In this paper, the problem of FDI and FTC of the wind turbine benchmark (Odgaard et al., 2013) has been addressed. The FDI scheme is based on the use of interval observers, that consider an unknown but bounded description of the noise and modeling errors, for fault detection and a row-reasoning approach for fault isolation. The proposed FTC scheme relies on the use of virtual sensors/actuators that are designed using an input/output model formulation and the model matching principle. A fault estimation scheme is also proposed based on batch least squares approach. The performance of the proposed FDI and FTC schemes has been assessed using the fault scenarios considered in the wind turbine benchmark. Satisfactory results have been obtained for both FDI and FTC schemes.

## Acknowledgements

This work has been funded by the Spanish MINECO through the project CY-CYT SHERECS (ref. DPI2011-26243), by the European Commission through contract i-Sense (ref. FP7-ICT-2009-6-270428), by UPC through the grant FPI-UPC E-01104 and by AGAUR through the contract FI-DGR 2013FIB00218.

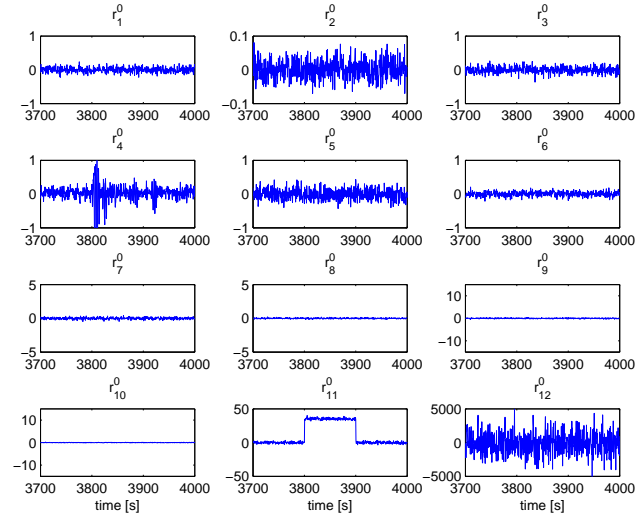


Figure 18: Nominal residuals  $r_1^0, \dots, r_{12}^0$  in *fault scenario 8*.

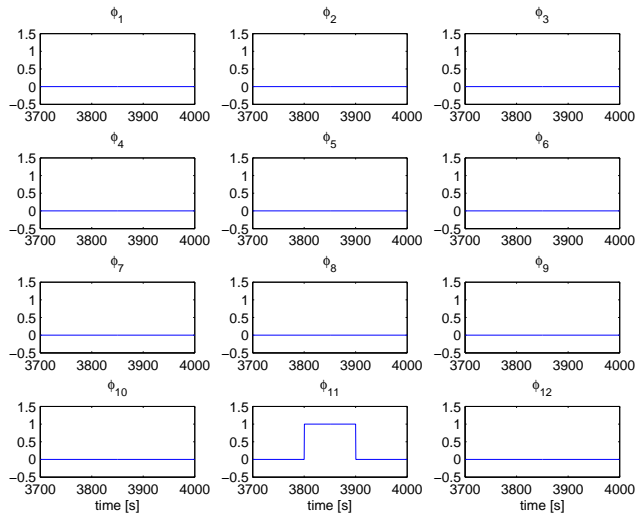


Figure 19: Observed fault signature components in *fault scenario 8*.

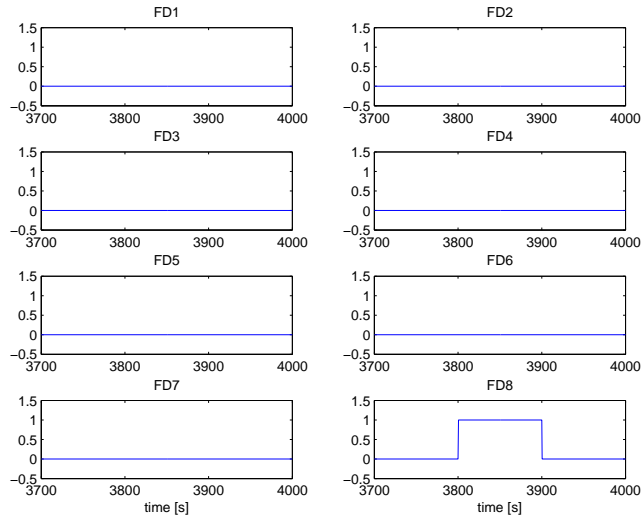


Figure 20: Fault detection results in *fault scenario 8*.

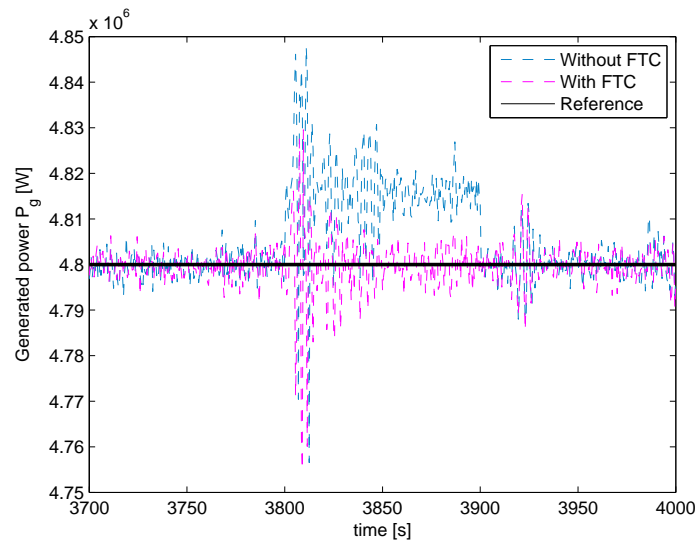


Figure 21: Generated power  $P_g$  without and with FTC in *fault scenario 8*.



- Alamo, T., Bravo, J., & Camacho, E. (2005). Guaranteed State Estimation by Zonotopes. *Automatica*, *41*, 1035–1043.
- Badihi, H., Zhang, Y., & Hong, H. (2013). Fuzzy gain-scheduled active fault-tolerant control of a wind turbine. *Journal of the Franklin Institute*, (in press).
- Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control*. Springer-Verlag Berlin Heidelberg.
- Blanke, M., & Lorentz, T. (2006). Satool: A software tool for structural analysis of complex automation systems. In *6th IFAC SAFEPROCESS* (pp. 673–678).
- Blesa, J., Puig, V., Romera, J., & Saludes, J. (2011). Fault diagnosis of wind turbines using a set-membership approach. In *Proceedings of the 18th IFAC World Congress* (pp. 8316–8321). Milan, Italy.
- Bossanyi, E. A. (2003). Wind turbine control for load reduction. *Wind Energy*, *6*, 229–244.
- Casau, P., Rosa, P., Tabatabaeipour, S., & Silvestre, C. (2012). Fault detection and isolation and fault tolerant control of wind turbines using set-valued observers. In *Proceedings of the 7th Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS 2012)* (pp. 120–125).
- Chen, C. T. (1984). *Linear systems theory and design*. New York, NY: Holt, Rinehart and Winston.
- Chen, W., Ding, S. X., Haghani, A., Naik, A., Khan, A. Q., & Yin, S. (2011). Observer-based FDI schemes for Wind Turbine Benchmark. In *Proceedings of the 18th IFAC World Congress* (pp. 7073–7078). Milan, Italy.
- Chilali, M., & Gahinet, P. (1996).  $H_\infty$  Design with pole placement constraints: an LMI approach. *IEEE Transactions on Automatic Control*, *41*, 358–367.
- Cordier, M., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M., & Travée-Massuyès, L. (2004). Conflicts versus analytical redundancy relations: A comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, *34*, 2163–2177.

- Dalai, M., Weyer, E., & Campi, M. (2005). Parametric identification of nonlinear systems: guaranteed confidence regions. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference* (pp. 6418–6423).
- Dobrila, C., & Stefansen, R. (2007). *Fault tolerant wind turbine control*. Master's thesis Technical University of Denmark, Kgl. Lyngby, Denmark.
- Edwards, C., Spurgeon, S. K., & Patton, R. J. (2000). Sliding mode observers for fault detection and isolation. *Automatica*, *36*, 541–553.
- Gertler, J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker.
- Isermann, R. (2006). *Fault Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, New York.
- Jain, T., Yamé, J. J., & Sauter, D. (2013). A novel approach to real-time fault accommodation in NREL's 5-MW wind turbine systems. *IEEE Transactions on Sustainable Energy*, *4*, 1082–1090.
- Jiang, J., & Zhang, Y. (2004). A novel variable-length sliding window block-wise least-squares algorithm for on-line estimation of time-varying parameters. *International Journal of Adaptive Control and Signal Processing*, *18*, 505–521.
- Kaczorek, T. (1982). Polynomial equation approach to exact model matching problem in multivariable linear systems. *International Journal of Control*, *36*, 531–539.
- Kamal, E., Aitouche, A., Ghorbani, R., & Bayart, M. (2012). Robust fuzzy fault-tolerant control of wind energy conversion systems subject to sensor faults. *IEEE Transactions on Sustainable Energy*, *3*, 231–241.
- Kendall, M., & Stuart, A. (1979). *The Advanced Theory of Statistics* volume 2. Charles Griffin.
- Ljung, L. (1987). *System Identification: Theory for the User*. Prentice Hall Information and System Sciences Series.
- Lunze, J., Rowe-serrano, D., & Steffen, T. (2003). Control Reconfiguration Demonstrated at a Two-Degrees-of-Freedom Helicopter Model. Cambridge, UK.

- Lunze, J., & Steffen, T. (2003). Control reconfiguration by means of a virtual actuator. In *Proceedings of the 5th Symposium on Fault Detection and Safety for Technical Processes (SAFEPROCESS03)* (pp. 133–138).
- Merritt, H. E. (1967). *Hydraulic control systems*. John Wiley and Sons, Inc.
- Meseguer, J., Puig, V., Escobet, T., & Saludes, J. (2010). Observer gain effect in linear interval observer-based fault detection. *Journal of Process Control*, 20, 944–956.
- Milanese, M., Norton, J., Piet-Lahanier, H., & Walter, É. (Eds.) (1996). *Bounding Approaches to System Identification*. Springer.
- Montes de Oca, S., Rotondo, D., Nejjari, F., & Puig, V. (2011). Fault Estimation and Virtual Sensor FTC Approach for LPV Systems. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference* (pp. 2251–2256). Orlando, Florida, USA.
- Odgaard, P., & Stoustrup, J. (2012a). Fault Tolerant Control of Wind Turbines using Unknown Input Observers. In *Proceedings of The 8th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Processes*.
- Odgaard, P., Stoustrup, J., Nielsen, R., & Damgaard, C. (2009). Observer based detection of sensor faults in wind turbines. In *Proceedings of the European Wind Energy Conference*.
- Odgaard, P. F., & Stoustrup, J. (2012b). Results of a Wind Turbine FDI Competition. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS)* (pp. 102–107). Mexico City, Mexico.
- Odgaard, P. F., Stoustrup, J., & Kinnaert, M. (2013). Fault-tolerant control of wind turbines: a benchmark model. *IEEE Transactions on Control Systems Technology*, 21, 1168–1182.
- Patton, R. J., & Klinkhieo, S. (2009). Actuator Fault Estimation and Compensation based on an Augmented State Observer Approach. In *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*. Shanghai, China.
- Ploix, S., Adrot, O., & Ragot, J. (1999). Parameter Uncertainty Computation in Static Linear Models. (pp. 1916–1921). volume 2.

- Puig, V., Montes de Oca, S., & Blesa, J. (2013). 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, 873–901.
- Puig, V., Saludes, J., & Quevedo, J. (2003). Worst-Case Simulation of Discrete Linear Time-Invariant Interval Dynamic Systems. *Reliable Computing*, 9, 251–290.
- Puig, V., Stancu, A., Escobet, T., Nejjari, F., Quevedo, J., & Patton, R. J. (2006). Passive Robust Fault Detection using Interval Observers: Application to the DAMADICS Benchmark Problem. *Control Engineering Practice*, 14, 621–633.
- Razavi-Far, R., & Kinnaert, M. (2013). A multiple observers and dynamic weighting ensembles scheme for diagnosing new class faults in wind turbines. *Control Engineering Practice*, 21, 1165–1177.
- Reinelt, W., Garulli, A., & Ljung, L. (2002). Comparing different approaches to model error modeling in robust identification. *Automatica*, 38, 787–803.
- Rotondo, D., Nejjari, F., Puig, V., & Blesa, J. (2012). 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.
- Sami, M., & Patton, R. (2012). Wind Turbine Power Maximisation Based On Adaptive Sensor Fault Tolerant Sliding Mode Control. In *Proceedings of the 2012 20th Mediterranean Conference on Control and Automation (MED12)* (pp. 1177–1182).
- Simani, S., & Castaldi, P. (2013). Active actuator fault-tolerant control of a wind turbine benchmark model. *International Journal of Robust and Nonlinear Control*, (in press).
- Sloth, C., Esbensen, T., & Stoustrup, J. (2010). Active and passive fault-tolerant LPV control of wind turbines. In *Proceedings of the American Control Conference* (pp. 4640–4646). Baltimore, USA.
- Sloth, C., Esbensen, T., & Stoustrup, J. (2011). Robust and Fault-Tolerant Linear Parameter-Varying Control of Wind Turbines. *Mechatronics*, 21, 645–659.

- Svärd, C., & Nyberg, M. (2012). Automated design of an fdi system for the wind turbine benchmark. *Journal of Control Science and Engineering*, (pp. 1–13).
- Tabatabaeipour, S. M., Odgaard, P. F., Bak, T., & Stoustrup, J. (2012). Fault Detection of Wind Turbines with Uncertain Parameters: A Set-Membership Approach. *Energies*, 5, 2424–2448.
- van der Veen, G. J., van Wingerden, J. W., Fleming, P. A., Scholbrock, A. K., & Verhaegen, M. (2013). Global data-driven modeling of wind turbines in the presence of turbulence. *Control Engineering Practice*, 21, 441–454.
- Wang, H., & Daley, S. (1996). Actuator fault diagnosis: An adaptive observer-based technique. *IEEE Transactions on Automatic Control*, 41, 1073–1078.
- Wei, X., Verhaegen, M., & van den Engelen, T. (2008). Sensor fault diagnosis of wind turbines for fault tolerant. In *Proceedings of 17th IFAC World Congress*. Seoul, Korea.