



Fault Detection of Wind Turbines with Uncertain Parameters

A Set-Membership Approach

Tabatabaeipour, Seyed Mojtaba; Odgaard, Peter Fogh; Bak, Thomas; Stoustrup, Jakob

Published in:
Energies

DOI (link to publication from Publisher):
[10.3390/en5072424](https://doi.org/10.3390/en5072424)

Publication date:
2012

Document Version
Early version, also known as pre-print

[Link to publication from Aalborg University](#)

Citation for published version (APA):

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(7), 2224-2248.
<https://doi.org/10.3390/en5072424>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Article

Fault Detection of Wind Turbines with Uncertain Parameters: A Set-Membership Approach

Seyed Mojtaba Tabatabaeipour ^{1,*}, Peter F. Odgaard ², Thomas Bak ¹ and Jakob Stoustrup ¹

¹ Automation & Control, Department of Electronic Systems, Aalborg University, Aalborg DK-9220, Denmark; E-Mails: tba@es.aau.dk (T.B.); jakob@es.aau.dk (J.S.)

² kk-electronic a/s, Ikast 7430, Denmark; E-Mail: peodg@kk-electronic.com

* Author to whom correspondence should be addressed; E-Mail: smt@es.aau.dk; Tel.: +45-9940-8738.

Received: 27 April 2012; in revised form: 11 June 2012 / Accepted: 6 July 2012 /

Published: 13 July 2012

Abstract: In this paper a set-membership approach for fault detection of a benchmark wind turbine is proposed. The benchmark represents relevant fault scenarios in the control system, including sensor, actuator and system faults. In addition we also consider parameter uncertainties and uncertainties on the torque coefficient. High noise on the wind speed measurement, nonlinearities in the aerodynamic torque and uncertainties on the parameters make fault detection a challenging problem. We use an effective wind speed estimator to reduce the noise on the wind speed measurements. A set-membership approach is used generate a set that contains all states consistent with the past measurements and the given model of the wind turbine including uncertainties and noise. This set represents all possible states the system can be in if not faulty. If the current measurement is not consistent with this set, a fault is detected. For representation of these sets we use zonotopes and for modeling of uncertainties we use matrix zonotopes, which yields a computationally efficient algorithm. The method is applied to the wind turbine benchmark problem without and with uncertainties. The result demonstrates the effectiveness of the proposed method compared to other proposed methods applied to the same problem. An advantage of the proposed method is that there is no need for threshold design, and it does not produce positive false alarms. In the case where uncertainty on the torque lookup table is introduced, some faults are not detectable. Previous research has not addressed this uncertainty. The method proposed here requires equal or less detection time than previous results.

Keywords: fault detection; wind turbines; set-membership; zonotopes

1. Introduction

In recent years, there is an increasing attention to wind energy as one of the most promising and abundant sources for sustainable energy. Operation and maintenance (O & M) costs constitute an important part of the cost of wind energy. The costs can be reduced by developing wind turbines that require less scheduled and especially non-scheduled service and has less downtime due to failure. This is important especially for offshore wind farms where the cost associated with O & M is higher and where weather conditions may prevent maintenance for long periods of time. A remedy to reduce O & M costs is a solution that provides us the possibility of producing power, probably with some degradation in the performance, after a fault has occurred until the next scheduled service. An efficient method to reduce O & M costs is early and accurate fault detection and diagnosis (FDD).

In different studies on wind turbine failure analysis, it is shown that the control system with its corresponding instrumentation (sensors and actuators) is the subsystem that has the highest failure frequency and it causes the highest percentage of downtime [1]. This emphasizes the need for effective fault diagnosis and more generally for fault tolerant control (FTC) methods for the wind turbine. Such a control system should be able to detect and isolate faults when they occur; and then to re-design the controller by changing the control law or the control configuration—in the case of severe faults—such that the faulty turbine with the new controller can remain producing power and present an acceptable performance from the time that a fault occurs until the next scheduled service.

In recent years there has been some works investigating the problem of fault diagnosis and fault tolerant control of wind turbines, see [2]. Some early works have used condition monitoring techniques such as vibration analysis, oil analysis, acoustic monitoring, *etc.* For a review of condition monitoring techniques used for wind turbines see the review papers [3,4]. In [5], evolutionary algorithms and classification methods are used for detection of rotor imbalance and aerodynamics asymmetry faults. The same problem is studied in [6] by investigating statistical properties of wind speed and output power. A method for detection of excessive tower oscillations due to rotor imbalance is proposed in [7]. The method is able to discriminate between turbulence and/or wave and rotor imbalance. Because structural damages can cause disastrous damages to the system, monitoring structural health is of crucial importance. For a review of damage detection methods for structural health monitoring of wind turbines see [8] and references therein. In recent modern wind turbines, a doubly fed induction generator (DFIG) is widely used. In [9] a diagnosis method based on signature analysis of rotor modulating signals for DFIGs is proposed where electrical incipient faults are considered. A model-based method for sensor fault detection of a DFIG is proposed in [10]. In [11] an observer based fault detection for detecting sensor faults in the pitch system is proposed. An unknown input observer is used in [12] for sensor fault detection in the drive train. Poure *et al.* [13] investigates fault detection and fault tolerant control of the electrical conversion systems.

A benchmark model for fault detection, isolation and accommodation of wind turbines was proposed in [14]. The benchmark is based on a model of a generic three blade horizontal variable speed wind turbine with a full converter coupling and a rated power of 4.8 MW. The aim of the benchmark model is to provide a common ground to test and compare different methods for fault detection and accommodation of wind turbines. Recently, a number of papers have been published on fault detection and isolation of

the benchmark proposed in [14]. In [15] a solution based on Kalman filters and diagnostic observers for residual generation with generalized likelihood ratio test for detection decision is proposed. Solutions based on support vector machines are proposed in [16]. In [17] a model based method is used to generate residuals on which an up-down counter based scheme is used for detection and isolation. Svärd *et al.* [18] present a generic automated design method for fault detection and isolation on this benchmark. A scheme based on fault detection and isolation estimators is proposed in [19]. In [20] a set-theoretic method is proposed for this problem. Kiasi *et al.* [21] focus on generalized likelihood ratio and associated statistical fault detection tools. A hybrid approach is taken in [22]. In [23] a data driven approach is taken on the benchmark model. In [24] a detection and isolation scheme based on a combination of parity equations and robust residual filtering is proposed.

In this paper we propose a state space based set-membership fault detection method for fault detection of the benchmark model. In real applications, noise, uncertainties and model differences always exist. To guarantee the reliability and performance of a fault detection method, it is important to ensure that it is insensitive to uncertainties and noise but at the same time sensitive to faults. Such a fault detection method is called robust. Robust fault detection methods are broadly divided into two categories: residual signal based and set-membership based. In the robust residual signal based fault detection, fault detection is based on checking the value of a residual signal against a threshold. If the value of the residual signal is greater than the threshold, a fault is detected. Designing a residual signal which is sensitive to faults and insensitive to noise and uncertainties is usually a difficult and challenging problem. Some of the well known residual based approaches are unknown input observers [25], eigenstructure assignment [26], and structured parity equations [27]. In the set-membership approaches, the noise, disturbance and uncertainties on the parameters of the model are assumed to be unknown but bounded with a priori known bound. Then, a set of models is generated for the system. Whenever a measurement is not consistent with any of the members of this set, a fault is detected.

In the control literature these approaches are known as set-membership, or error bounded methods; see [28,29] for a review. An advantage of the set-membership approach is that there is no need for a threshold design. Set-membership approaches work in the state space or the parameter space. The set of states or parameters that are consistent with the measurements, the model of the system, and the bounds on the uncertainties, disturbance, and noise are calculated as a closed set. If the set of states or the parameters consistent with a new measurement is outside this set, a fault is detected. Another advantage of set-membership approaches is that, if the given bounds on the uncertainties, noise and disturbance are realistic, no positive false alarm is produced. The drawback of the method is its conservatism due to propagation of uncertainties and due to over-approximations required in the set computations.

All of the aforementioned methods for fault detection of the benchmark are residual signal based. Moreover, none of them take uncertainties on parameters into account. In [30] a set-membership approach is proposed, which investigates the application of the parameter estimation based method for fault detection of wind turbines. However, they assume that the system can be represented by a model that is linear in the estimated parameters. In this work, we use the state estimation set-membership method using the nonlinear model of the system. The nonlinear term is approximated by an interval at each sample time. While Blesa *et al.* [30] cannot detect fault No. 6 in the benchmark, we can detect this fault. Also, all of the faults are detected at the same time or sooner than the results reported in [30]. Moreover,

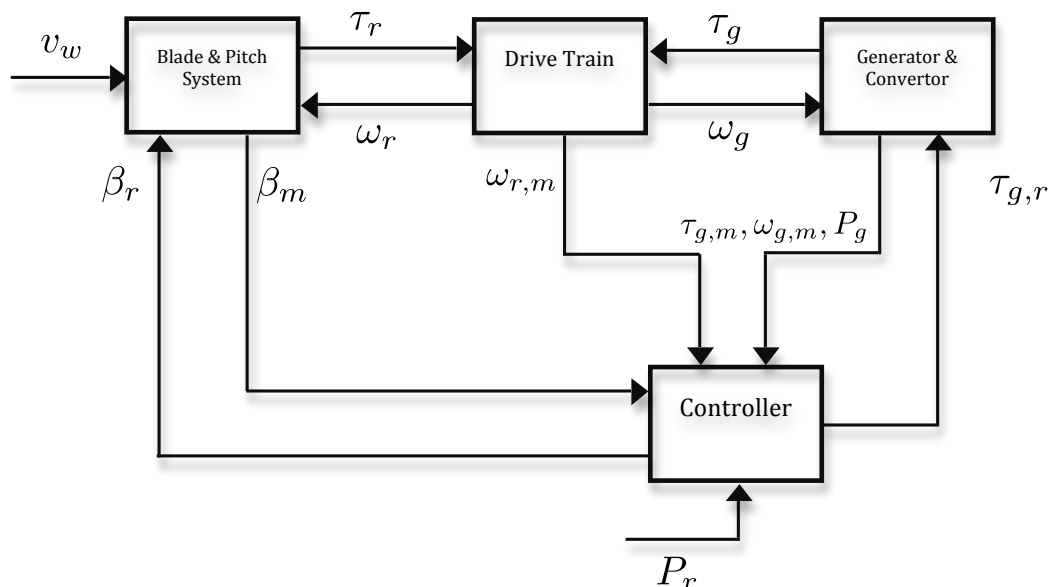
none of the aforementioned methods consider uncertainty on the model of the system. In our work, we consider uncertainties on the torque coefficient look-up table and also on the system matrices. To deal with the computational geometry of the proposed method we use zonotopes for set computations and to deal with the parametric uncertainties we use matrix zonotopes. As a result, all of the computations are performed on zonotopes, which offer low computational complexity and require low allocated memory. Hence, the proposed method is computationally efficient.

The paper is organized as follows. In Section 2, preliminaries and basic required definitions are given. The benchmark model and fault scenarios are explained in Section 3. In Section 4, the state space set-membership approach is given. By defining required operations on zonotopes, implementation of the proposed method using zonotopes is given in Section 5. In Section 6, we explain how we deal with uncertainties. Simulation results and discussion are given in Section 7. Finally, Section 8 concludes the paper.

2. Wind Turbine Description

A wind turbine converts a part of the kinetic energy of the wind into electrical energy. In this paper we consider a benchmark model of a three blade horizontal axis wind turbine with a full converter. The wind turbine model consists of four parts: blade and pitch systems, drive train, generator and converter, and the controller, see Figure 1. The rotational torque due to the airflow of the wind on the rotor blades rotates the rotor. As a result a part of wind energy is converted to mechanical energy. The drive train transmits this mechanical energy to the generator, and the generator converts the mechanical energy to electrical energy. To control the power, we can manipulate the blade pitch angle or the rotational speed of the rotor. The blade pitch is controlled through the pitch actuators. The rotational speed of the generator or the rotor is controlled by the generator torque. The generator is fully coupled with a converter, which provides us with the ability to control the generator torque. The rotor speed, the generator speed, and the pitch positions of all blades are measured with two sensors. In the following, different subsystems are described in more details.

Figure 1. Block diagram of the wind turbine model.



2.1. Blade and Pitch System

This model is comprised of two parts: the aerodynamic model and the pitch model.

Aerodynamic Model

The flow of the wind on each blade produces the drag and lift forces. The tangential component of these forces generates a tangential force which creates a rotational torque. The aerodynamic torque acting on the rotor is calculated by integrating the tangential torque along the blades length. The aerodynamic torque is given by (see [31,32]) :

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

where C_q is the torque coefficient, which is a function of the pitch angle β and the ratio between the speed of the blade tip and the wind speed, which is known as the tip-speed ratio and given by:

$$\lambda(t) = \frac{R \omega_r(t)}{v_w(t)} \quad (2)$$

where R is the radius of the blades, v_w is the wind speed and ω_r is the rotor speed. C_q is given in the form of a look-up table. The above equation assumes that the blades are controlled collectively and hence have the same pitch angle. When the blades have different pitch angles, the aerodynamic torque can be approximated by:

$$\tau_r(t) = \sum_{i=1}^3 \frac{1}{6} \rho \pi R^3 C_q(\lambda(t), \beta_i(t)) v_w^2 \quad (3)$$

Pitch System Model

The pitch actuator which rotates the blades is a hydraulic pitch system. The pitch actuator might be modeled by a first order system, see [31], but here we choose a second-order model to include the blade inertia into the account. The closed loop dynamic of the pitch systems is given by Merritt [33]:

$$\frac{\beta(s)}{\beta_r(s)} = \frac{\omega_n^2}{s^2 + 2\xi \omega_n s + \omega_n^2} \quad (4)$$

where β_r is the pitch reference.

2.2. Drive Train Model

The drive train consists of a low speed shaft and a high-speed shaft, which are interconnected through a transmission with a gear ration of N_g . It is modeled as two shafts with moments of inertia and frictional force that are connected with a gearing with efficiency of η_{dt} . The model of the drive train is depicted in Figure 2. The dynamics of the low-speed shaft is given by:

$$J_r \dot{\omega}_r = \tau_r - \tau_l - B_r \omega_r \quad (5)$$

where J_r, B_r denote the moment inertia and the viscous friction of the low speed shaft, ω_r is the rotational speed of the rotor, and τ_l is the torque on the low-speed shaft. Similarly, the dynamics of the low speed shaft is:

$$J_g \dot{\omega}_g = \tau_h - \tau_g - B_g \omega_g \quad (6)$$

where J_g, B_g denote the moment inertia and the viscous friction of the high speed shaft respectively, ω_g is the rotational speed of the generator, and τ_h is the torque on the high-speed shaft. The gearbox is modeled by:

$$\tau_h = \frac{\tau_l}{N_g} \tag{7}$$

The drive train torsion is modeled by a torsion spring and a friction coefficient as:

$$\tau_l = K_{dt}\theta_{\Delta} + \eta_{dt}B_{dt}\dot{\theta}_{\Delta} \tag{8}$$

where K_{dt} denotes the torsion stiffness of the drive train, B_{dt} is the torsion damping coefficient of the drive train, and $\theta_{\Delta}(t)$ denotes the torsion angle of the drive train which is equal to:

$$\theta_{\Delta} = \theta_r - \frac{\theta_g}{N_g} \tag{9}$$

where θ_r is the angle of the low speed shaft and θ_g is the angle of the high speed shaft. It is easy to see that by using the above equations, the overall dynamics of the drive train in the state space form can be re-written as:

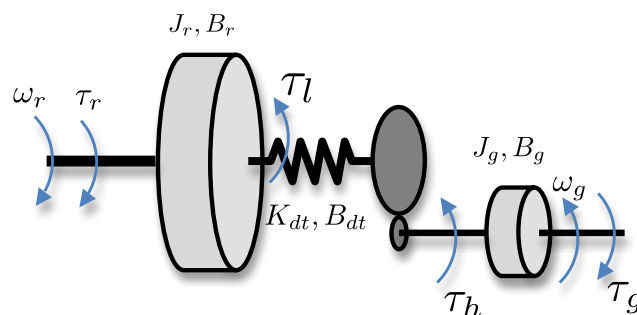
$$\begin{bmatrix} \dot{\omega}_r(t) \\ \dot{\omega}_g(t) \\ \dot{\theta}_{\Delta}(t) \end{bmatrix} = A_{DT} \begin{bmatrix} \omega_r(t) \\ \omega_g(t) \\ \theta_{\Delta}(t) \end{bmatrix} + B_{DT} \begin{bmatrix} \tau_r(t) \\ \tau_g(t) \end{bmatrix} \tag{10}$$

where

$$A_{DT} = \begin{bmatrix} -\frac{B_{dt}+B_r}{J_r} & \frac{B_{dt}}{N_g J_r} & -\frac{K_{dt}}{J_r} \\ \frac{\eta_{dt} B_{dt}}{N_g J_g} & -\frac{\eta_{dt} B_{dt} - B_g}{N_g^2 J_g} & \frac{\eta_{dt} K_{dt}}{N_g J_g} \\ 1 & -\frac{1}{N_g} & 0 \end{bmatrix} \tag{11}$$

$$B_{DT} = \begin{bmatrix} \frac{1}{J_r} & 0 \\ 0 & -\frac{1}{J_g} \\ 0 & 0 \end{bmatrix} \tag{12}$$

Figure 2. Schematic diagram of drive train.



2.3. Generator and Converter Models

The generator torque is controlled by a reference $\tau_{g,r}$. The dynamic of the converter is approximated by a first order system as:

$$\frac{\tau_g(s)}{\tau_{g,r}(s)} = \frac{\alpha_{gc}}{s + \alpha_{gc}} \tag{13}$$

The produced power by the generator is given by:

$$P_g(t) = \eta_g \omega_g(t) \tau_g(t) \quad (14)$$

where η_g is the efficiency of the generator [31] Model parameters can be found in [14].

2.4. Fault Scenarios

The following types of fault are considered in the benchmark: sensor faults, actuator faults, and system faults.

Sensor Faults

Sensor faults and their description are summarized in Table 1 where β_{j,m_i} represent the i th measurement of the j th pitch angle, $\omega_{r,m_1}, \omega_{r,m_2}$ are respectively the measurement from the first and the second rotor speed sensor, and $\omega_{g,m_1}, \omega_{g,m_2}$ are the measurement from the first and second generator speed sensor.

Table 1. List of sensor faults.

Fault No.	Fault	Period	Description
1	$\beta_{1,m_1} = 5^\circ$	2,000–2,100	Pitch sensor fixed value
2	$\beta_{1,m_2} = 1.2\beta_{1,m_1}$	2,300–2,400	Pitch sensor gain factor
3	$\beta_{3,m_1} = 10^\circ$	2,600–2,700	Pitch sensor fixed value
4	$\omega_{r,m_1} = 1.4 \text{ rad/s}$	1,500–1,600	Rotor speed fixed value
5	$\omega_{r,m_2} = 1.1\omega_{r,m_2}$ $\omega_{g,m_1} = 0.9\omega_{g,m_1}$	1,000–1,100	Rotor and generator speed gain factor

Actuator Faults

Two kinds of actuator faults are considered for the pitch system. The first fault is a hydraulic pressure drop in actuator 2. The pressure drop is modeled as an abrupt change of parameters of the pitch system from ω_n, ξ to ω_{n2}, ξ_2 . The second fault is an air content increase in the actuator oil. This fault is modeled as a gradual change in the parameters of the pitch actuator 3. The parameters are changed linearly from ω_n, ξ to ω_{n3}, ξ_3 in a period of 30 s, remain active for 40 s, and decrease again for 30 s. Actuator faults are summarized in Table 2.

Table 2. List of actuator faults.

Fault No.	Fault	Period	Description
6	abrupt change of $\omega_n, \xi \rightarrow \omega_{n2}, \xi_2$	2,900–3,000	pressure drop
7	slow change of $\omega_n, \xi \rightarrow \omega_{n3}, \xi_3$	3,400–3,500	air content increase in the oil

System Faults

A system fault is considered in the benchmark. The system fault is a bias in the generator control loop which yields a bias in the generator torque: $\tau_g = \tau_g + 2000 Nm$. Table 3 summarizes the system fault.

Table 3. System fault.

Fault No.	Fault	Period	Description
8	$\tau_g = \tau_g + 2000 Nm$	3,800–3,900	Offset

The wind turbine with fault scenarios are simulated using Simulink. The Simulink model is publicly available, see [14]. For simulation of a fault scenario, a fault signal is defined, which is 1 for the time period in which the fault occurs and zero if otherwise. This fault signal is used in the Simulink model. For each fault scenarios when the corresponding fault signal is 1, the parameters of the Simulink model switches to the parameters of the corresponding faulty model.

3. State Space Set-Membership Fault Detection

Consider the following nonlinear model:

$$x(k + 1) = f(x(k), u(k), w(k)) \tag{15}$$

$$y(k) = g(x(k), v(k)) \tag{16}$$

where $x(k) \in \mathbb{R}^n$ is the state, $y(k) \in \mathbb{R}^m$ is the output, $u(k) \in \mathbb{R}^p$ is the input, $w(k) \in \mathbb{R}^n$ is disturbance and $v(k) \in \mathbb{R}^m$ is noise. It is assumed that the noise and disturbance are unknown but bounded *i.e.*, $w(k) \in \mathcal{W}$ and $v(k) \in \mathcal{V}$. Moreover, it is assumed that the initial condition is given in a compact set $x(0) \in \mathcal{X}_0$.

The set-membership approach uses the consistency principle for fault detection. Each iteration calculates the consistency of the current measurement with a set, $\mathbb{X}^c(k)$, that contains all of the states that are consistent with a given model of the system, the initial condition set, the bounds on the disturbance and noise, and the input-output sequence up to the current sample time. If the current measurement is not consistent with this set, a fault is detected. Computation of $\mathbb{X}^c(k)$ consists of two steps: a prediction step and a correction step. At the prediction step, having $\mathbb{X}^c(k - 1)$, the predicted set $\mathbb{X}^p(k)$ is defined as:

$$\begin{aligned} \mathbb{X}^p(k) = \{ & f(x(k - 1), u(k - 1), w(k - 1)) : x(k - 1) \in \mathbb{X}^c(k - 1) \\ & u(k - 1) = u(k - 1), w(k - 1) \in \mathcal{W} \} \end{aligned} \tag{17}$$

The predicted set is one-step ahead prediction of $\mathbb{X}^c(k - 1)$. This set is then corrected using the information provided by the current measurement $y(k)$. Given the current measurement $y(k)$, the set of all states that are consistent with it are given by:

$$\mathbb{X}^y(k) = \{x \in \mathbb{R}^n : \exists v \in \mathcal{V} \text{ such that } g(x, v) = y(k)\} \tag{18}$$

When noise is additive *i.e.*, $g(x, v) = h(x) + v$, then:

$$\mathbb{X}^y(k) = \{x : h(x) \in y(k) \oplus (-V)\} \tag{19}$$

where \oplus denotes the Minkowski sum which is defined as follows. Given two sets $\mathcal{X}, \mathcal{Y}, \mathcal{X} \oplus \mathcal{Y} = \{x+y | x \in \mathcal{X}, y \in \mathcal{Y}\}$. Then, the corrected set is defined as the intersection of the set of states consistent with the output and the predicted set:

$$\mathbb{X}^c(k) = \mathbb{X}^p(k) \cap \mathbb{X}^y(k) \tag{20}$$

A fault is detected if this set is empty:

$$\mathbb{X}^p(k) \cap \mathbb{X}^y(k) = \emptyset \tag{21}$$

The overall algorithm for fault detection is given in Table 4.

Table 4. State space set-membership fault detection.

Algorithm 1

Given $f, g, \mathcal{X}_0, \mathcal{V}, \mathcal{W}$

$k = 0, \text{Fault} \leftarrow \text{False}, \mathbb{X}^c(k) = \mathcal{X}_0$

While $\text{Fault} = \text{False}$ $k = k + 1$

Given $u(k)$, find the prediction set:

$\mathbb{X}^p(k) = \{f(x(k-1), u(k-1), w(k-1)) : x(k-1) \in \mathbb{X}^c(k-1)$

$u(k-1) = u(k-1), w(k-1) \in \mathcal{W}\}$

Given $y(k)$, find $\mathbb{X}^y(k)$:

$\mathbb{X}^y(k) = \{x \in \mathbb{R}^n : \exists v \in \mathcal{V}, g(x, v) = y(k)\}$

$\mathbb{X}^c(k) = \mathbb{X}^p(k) \cap \mathbb{X}^y(k)$

If $\mathbb{X}^c(k) = \emptyset$, then $\text{Fault} = \text{True}$

end

Algorithm 1 is in a very general form. Exact computation of the corrected and the predicted sets using the given general nonlinear model is, if not impossible, very difficult. Therefore, in the literature different authors have assumed some simplifying assumptions about the nonlinear model. In the following section, we explain how we use interval analysis to find an interval approximation of the nonlinear aerodynamics to adapt the algorithm for wind turbine fault detection.

3.1. Fault Detection of The Wind Turbine

The model of the drive train and the converter can be summarized as follows:

$$x(k+1) = \begin{bmatrix} \omega_r(k+1) \\ \omega_g(k+1) \\ \theta_\Delta(k+1) \\ \tau_g(k+1) \\ \beta(k+1) \\ \beta(k+2) \end{bmatrix} = A \begin{bmatrix} \omega_r(k) \\ \omega_g(k) \\ \theta_\Delta(k) \\ \tau_g(k) \\ \beta(k) \\ \beta(k+1) \end{bmatrix} + B \begin{bmatrix} \tau_{g,r}(k) & \beta_r(k) \end{bmatrix}^T + B_{\tau_r} \tau_r(k) \tag{22}$$

$$y(k) = Cx + v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0_{1 \times 3} & 0_{1 \times 3} \\ 0 & 1 & 0 & 0 & 0_{1 \times 3} & 0_{1 \times 3} \\ 0 & 0 & 0 & 1 & 0_{1 \times 3} & 0_{1 \times 3} \\ 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} & I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} x(k) + v \tag{23}$$

where A and B are produced by an appropriate augmentation of the corresponding matrices of the drive train, the generator and the pitch system. The notation $(\cdot)_{m \times n}$ denotes a m by n matrix. Assume that at $k - 1$, we have $\mathbb{X}^c(k - 1)$. Then, we use Equation (22) to find $\mathbb{X}^p(k)$. If we can find an interval approximation of $\tau_r(k)$, then we have:

$$\mathbb{X}(k)^p = A\mathbb{X}^c(k - 1) \oplus B\tau_{g,r}(k - 1) \oplus B_{\tau_r}[\tau_r(k - 1)], \tag{24}$$

where $[a] = [\underline{a}, \bar{a}]$ denotes a closed interval which is the set $[a, \bar{a}] = \{x \in \mathbb{R} : \underline{a} \leq x \leq \bar{a}\}$. Moreover, having the measurement $y(k)$ and assuming that \mathcal{V} is given as the box $\mathcal{V} = ([v_{\omega_r}], [v_{\omega_g}][v_{\tau_r}])$, then the set, $[x_{dc}(k)]^y$ is obtained as:

$$\begin{aligned} [x_{dc}(k)]^y &= \{ \omega_r(k) - \underline{v}_{\omega_r} \leq \omega_r \leq \omega_r(k) + \bar{v}_{\omega_r} \\ &\quad \omega_g(k) - \underline{v}_{\omega_g} \leq \omega_g \leq \omega_g(k) + \bar{v}_{\omega_g} \\ &\quad \tau_g(k) - \underline{v}_{\tau_g} \leq \tau_g \leq \tau_g(k) + \bar{v}_{\tau_g} \} \end{aligned} \tag{25}$$

In the following, we show how we can find an interval approximation of $\tau_r(k)$ as $[\tau_r(k)] = [\underline{\tau}_r(k), \bar{\tau}_r(k)]$. Using interval analysis we have:

$$\bar{\tau}_r(k) = \sum_{i=1}^3 \frac{1}{6} \rho \pi \overline{C_{qi}}(k) \bar{v}_w^2(k) \tag{26}$$

$$\underline{\tau}_r(k) = \sum_{i=1}^3 \frac{1}{6} \rho \pi \underline{C_{qi}}(k) \underline{v}_w^2(k) \tag{27}$$

C_q is given in the form of a look-up table as a function of λ and β . We have $[\lambda(k)] = [\underline{\lambda}(k), \bar{\lambda}(k)] = [\frac{R\omega_r}{\underline{v}_w}, \frac{R\bar{\omega}_r}{\bar{v}_w}]$ and $[\beta_i]$ can be computed using a set-membership observer for the pitch system. Then:

$$\overline{C_{qi}} = \max(C_q([\lambda(k)], [\beta_i(k)])) \tag{28}$$

$$\underline{C_{qi}} = \min(C_q([\lambda(k)], [\beta_i(k)])) \tag{29}$$

Note that using this method the uncertainties on the C_q table can also be handled easily. In order to find $[v_w]$, we need a bound on the measurement noise for the wind speed. The wind speed measurement has a high level of noise and there is a possibility of offset on the sensor. Therefore, using information from this sensor would result in a very conservative over-approximation of $[\tau_r]$ which in turn would yield a poor performance of the fault detection algorithm. To address this problem we use an estimation of the effective wind speed instead of the measurement from the hub.

There are different methods to estimate effective wind speed, see [34,35]. We use the method suggested in [34]. The proposed wind speed estimator consists of three parts: a state estimator, an input estimator and a look-up table. The state estimator estimates the states of the drive train. The input estimator which is a PI controller estimates the aerodynamic torque which is fed to the state estimator.

Estimates of the rotor speed, the pitch angle from the state estimator and the aerodynamic torque are used as input to the look-up table to calculate the effective wind speed. The block diagram of the effective wind speed estimator is shown in Figure 3. We assume a bound on the estimation error of the EWS denoted by e_{vw} . Figure 4 shows the interval approximation of the aerodynamic torque using EWS estimation and its real value for the time period from 400 s to 900 s.

Figure 3. Block diagram of the effective wind speed estimator.

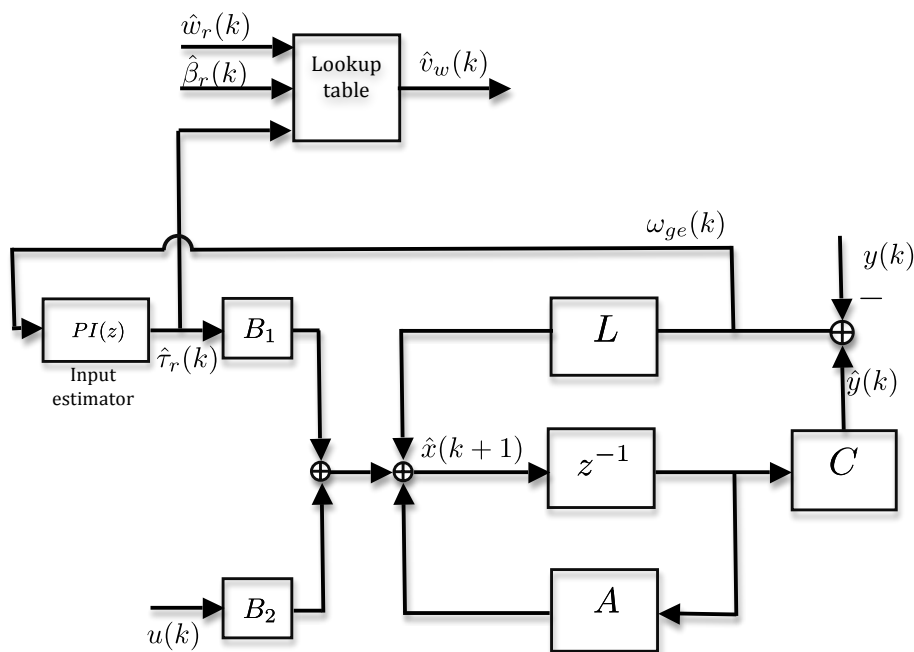
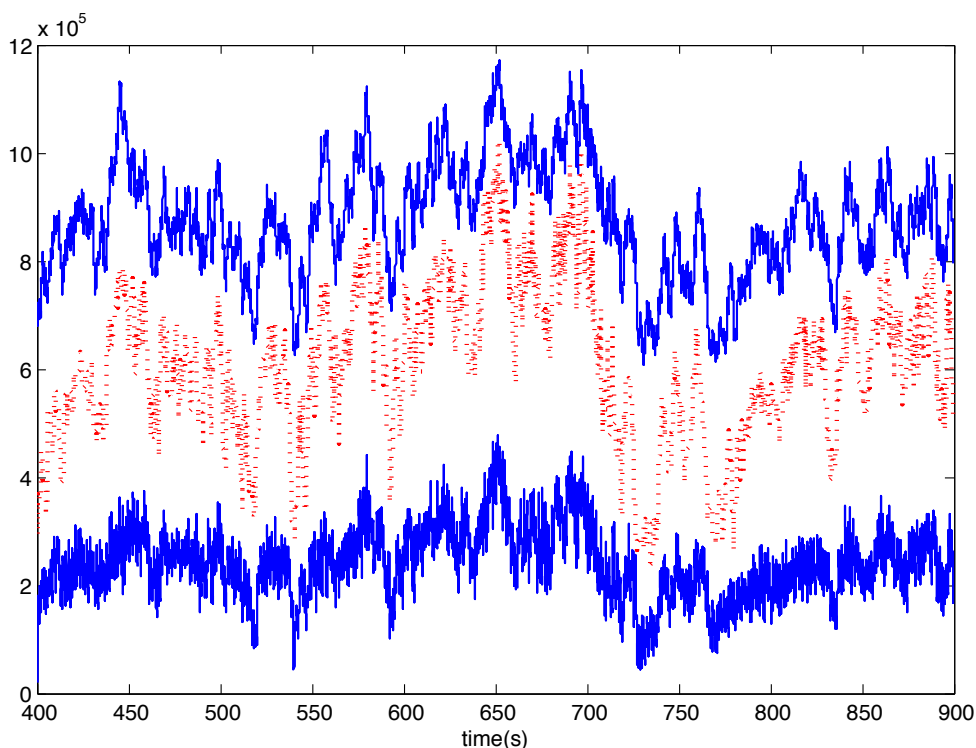


Figure 4. Interval approximation of τ_r : τ_r dotted line, $\overline{\tau_r}, \underline{\tau_r}$: dashed line.



The overall algorithm for fault detection is given in Table 5. To avoid computational complexity, at each step k , $\mathbb{X}^p(k+1)$ can be over approximated by a bounding set denoted by $\square\mathbb{X}^p(k+1)$.

Table 5. State space set-membership fault detection.

Algorithm 2

1. **Given** $A, B, \mathcal{X}_0, \mathcal{V}, e_{v_w}$
2. $k = 1$, Fault \leftarrow False, $\mathbb{X}^p(k) = \mathcal{X}_0$
3. **While** Fault=False
 - 3.1. Get input-output data $\{u(k), y(k)\}$
 - 3.2 Find $\mathbb{X}^y(k)$ using 21
 - 3.3. $\mathbb{X}^c(k) = \mathbb{X}^y(k) \cap \mathbb{X}^p(k)$
 - 3.4. Get $v_w(k)$ from EWS estimator
 - 3.5. $\overline{v_w}(k) = v_w(k) + e_{v_w}, \underline{v_w}(k) = v_w(k) - e_{v_w}$
 - 3.6. $[\lambda(k)] = \left[\frac{R\omega_r(k)}{v_w(k)}, \frac{R\omega_r(k)}{v_w(k)} \right]$
 - 3.7. $\overline{C_{qi}} = \max(C_q([\lambda(k)], [\beta_i(k)]))$
 - 3.8. $\underline{C_{qi}} = \min(C_q([\lambda(k)], [\beta_i(k)]))$
 - 3.9. $\overline{\tau_r}(k) = \sum_{i=1}^3 \frac{1}{6} \rho \pi \overline{C_{qi}}(k) \overline{v_w}^2(k)$
 - 3.10. $\underline{\tau_r}(k) = \sum_{i=1}^3 \frac{1}{6} \rho \pi \underline{C_{qi}}(k) \underline{v_w}^2(k)$
 - 3.11. $[\tau_r(k)] = [\underline{\tau_r}(k), \overline{\tau_r}(k)]$
 - 3.12. $\mathbb{X}^p(k+1) = A\mathbb{X}^c(k) \oplus Bu(k) \oplus B_{\tau_r}[\tau_r(k)]$
 - 3.13. $\mathbb{X}^p(k+1) \leftarrow \square\mathbb{X}^p(k+1)$
 - 3.14. If $\mathbb{X}^c(k) = \emptyset$, then Fault=True

end

4. Implementation Using Zonotopes

To perform the set computations in Algorithm 2, a specific set representation must be used. To deal with the underlying computational geometry required by the algorithm, different representation of these sets are used. These representations include ellipsoids, polyhedrons, parallelotopes, intervals, or zonotopes, see [36] and references therein. The representation must be efficient with respect to the set operations required in the algorithm. These operations are affine transformation, Minkowski sum, and intersection. The benefit of using ellipsoidal sets is their simplicity, but they are not closed under Minkowski sum and intersection. Polytopes are closed under Minkowski sum and intersection. When uncertainty is included, which is the case in this paper, convex hull computation is also necessary. Polytopes are also closed under convex hull computation and they provide accurate results but they suffer from high computational complexity. Minkowski sum and convex hull computation for polytopes are in general restricted to systems with a maximum of 4–6 states. Zonotopes are closed under affine transformation and Minkowski sum and they offer low time and memory complexity. When uncertainty is included, we propose using zonotope matrices which eliminates the need for convex hull computation. For a system with dimension n , computational complexity is $\mathcal{O}(n^3)$ for linear transformation of a zonotope and is $\mathcal{O}(n)$ for the Minkowski sum of two zonotopes. For intersection, we

use a computationally efficient method that over-approximates the intersection of the output consistent set with the prediction set by a zonotope. As a result, all operations are performed with a low computational complexity. In the following, after introducing zonotopes, we explain the implementation of Algorithm 2, using zonotopes.

Zonotopes are a special class of convex polytopes. A zonotope of order m in \mathbb{R}^n is an affine image of a m -dimensional unitary box \mathbf{B}^m in \mathbb{R}^n . Given the vector $p \in \mathbb{R}^n$, and the matrix $H \in \mathbb{R}^{n \times m}$, then the set

$$p \oplus H\mathbf{B}^m = \{p + Hz | z \in \mathbf{B}^m\} \tag{30}$$

is a zonotope which is the affine image of \mathbf{B}^m defined by p, H . Here, p is the center of zonotope. A zonotope can also be considered as the Minkowski sum of a finite number of line segments. In this case the zonotope is represented by:

$$Z = \{z \in \mathbb{R}^n | z = c + \sum_{i=1}^p x_i g_i, -1 \leq x_i \leq 1\} \tag{31}$$

here, c is the center of zonotope and g_i 's are called generators. Therefore, the zonotope $Z = p \oplus H\mathbf{B}^m$ is actually the Minkowski sum of the line segments defined by columns of H centered on p . The operations required to implement the Algorithm 2 are the following:

- Minkowski sum of two zonotopes
- Linear mapping of a zonotope
- Calculating the intersection of a strip and a zonotope

4.1. Minkowski Sum of Two Zonotopes

To calculate Minkowski sum of two zonotopes we use the following property.

Property 1 Given two zonotopes $Z_1 = p_1 + H_1\mathbf{B}_1^m$ and $Z_2 = p_2 + H_2\mathbf{B}_2^m$, then the Minkowski sum of them is also a zonotope and we have:

$$Z_1 \oplus Z_2 = (p_1 + p_2) \oplus \begin{bmatrix} H_1 & H_2 \end{bmatrix} \mathbf{B}^{m_1+m_2} \tag{32}$$

In other words, the Minkowski sum of two zonotopes is obtained by addition of their centers and concatenation of their generators. This shows the simplicity of this operation on zonotopes.

4.2. Linear Mapping of a Zonotope

The following property is used to find a linear image of a zonotope.

Property 2 Given a zonotope $Z = p + H\mathbf{B}^m$ and a linear map L , the image of Z by L is a zonotope given by:

$$LZ = Lp \oplus \begin{bmatrix} LH \end{bmatrix} \mathbf{B}^m \tag{33}$$

4.3. Computing the Intersection of a Zonotope and a Strip

In step 3.4 the intersection of consistent states with the measurement and the prediction set must be computed. The set of consistent states with the measurement, $\mathbb{X}^y(k)$, is actually the intersection of q strips where q is the number of rows of the output matrix C . Given a measurement $y(k)$, the set of states that are consistent with it is given by:

$$\mathbb{X}^y(k) = \{x \in \mathbb{R}^n \mid |Cx(k) - y(k)| \leq v\} \tag{34}$$

which is the intersection of q strips:

$$\mathbb{X}^y(k) = \bigcap_{i=1}^q \mathbb{X}_{y_i}(k) \tag{35}$$

where $\mathbb{X}_{y_i}(k)$ is the strip that contains the set of states consistent with the i th component of the measurement $y_i(k)$, which is:

$$\mathbb{X}_{y_i}(k) = \{x \in \mathbb{R}^n \mid |c_i x - y_i(k)| \leq v_i\} \tag{36}$$

where c_i is the i th row of C and v_i shows the i th element of v . Therefore,

$$\mathbb{X}^c(k) = \mathbb{X}^y(k) \cap \mathbb{X}^p(k) \subseteq \left(\bigcap_{i=1}^q \mathbb{X}_{y_i}(k)\right) \cap \mathbb{X}^p(k) \tag{37}$$

Therefore, to calculate $\mathbb{X}^y(k) \cap \mathbb{X}^p(k)$, we need sequential calculation of intersection of a zonotope and a strip such that at each time the intersection is over-approximated by a zonotope. This is because intersection of a zonotope and a strip is not necessarily a zonotope. In the following we explain how we can perform this over-approximation. But, before computing the intersection $\mathbb{X}^y(k)$ and $\mathbb{X}^p(k)$, we need to check if they intersect.

Checking consistency of a zonotope and a strip: To check if a zonotope and a strip intersect we use the following definition.

Definition 1 [37] Given a zonotope $Z = p \oplus HB^m$ and a strip $S = \{x \in \mathbb{R}^n \mid |cx - d| \leq \sigma\}$, the zonotope support strip is defined by

$$F_S = \{x \in \mathbb{R}^n \mid q_d \leq cx \leq q_u\} \tag{38}$$

where q_u and q_d are defined as:

$$q_u = \max_{x \in Z} cx \tag{39}$$

$$q_d = \min_{x \in Z} cx \tag{40}$$

which are calculated by:

$$q_u = cp + \|H^T c\|_1 \tag{41}$$

$$q_d = cp - \|H^T c\|_1 \tag{42}$$

where $\|\cdot\|_1$ is the 1-norm of a vector. Then, $S \cap Z = \emptyset$ if and only if:

$$q_u < \frac{d}{\sigma} - 1 \text{ or } q_d > \frac{d}{\sigma} + 1 \tag{43}$$

Intersection of a zonotope and a strip: Given a zonotope and a strip, the following property gives a family of zonotopes parameterized by the vector λ that over-approximates the intersection of the zonotope and the strip.

Property 3 [36] *Given the zonotope $Z = p \oplus HB^r \subset \mathbb{R}^n$, the strip $S = \{x \in \mathbb{R}^n \mid |cx - d| \leq \sigma\}$ and the vector $\lambda \in \mathbb{R}^n$. Define:*

$$\hat{p}(\lambda) = p + \lambda(d - cp) \tag{44}$$

$$\hat{H}(\lambda) = \begin{bmatrix} (I - \lambda c)H & \sigma\lambda \end{bmatrix} \tag{45}$$

Then $S \cap Z \subseteq \hat{p}(\lambda) + \hat{H}(\lambda)\mathbf{B}^{r+1}$.

The above over-approximation might not be a tight approximation. To find a tight over-approximation, λ must be chosen such that an approximation criterion is minimized. In [36] two approaches are proposed. In the first approach the segments of zonotopes are minimized by means of minimizing the Frobenius norm of $\hat{H}(\lambda)$. The λ that minimizes the Frobenius norm of $\hat{H}(\lambda)$ is given by:

$$\lambda^* = \frac{HH^T c^T}{cHH^T c^T + \sigma^2} \tag{46}$$

The advantage of this approach is its computational simplicity. If a more accurate result is required, a possible solution is to choose the λ that minimizes the volume of the intersection[36]. However, this requires solving a convex optimization problem at each iteration, which results in more computational complexity. In our simulations we have used the segment minimization approach.

5. Adding Uncertainty

Two kinds of uncertainties are considered in this work. Uncertainties on the look-up table of the torque coefficient and parametric uncertainties on the system matrices. The former can be handled very easily as explained before, and in the following we will explain how to deal with the latter. There are several methods to represent parametric uncertainties on system matrices, including the use of matrix polytopes. The problem with matrix polytopes is that the multiplication of a matrix polytope and a zonotope is no longer a zonotope. Also, over-approximating a polytope with a zonotope is computationally demanding [38]. In this work we propose using matrix zonotopes. The advantage of using matrix zonotopes is that the multiplication of a matrix zonotope and a zonotope is a zonotope.

A matrix zonotope is defined by:

$$A^{[z]} = \{A^0 + \sum_{i=1}^k x_i A^i \mid -1 \leq x_i \leq 1, A^i \in \mathbb{R}^{n \times n}\} \tag{47}$$

The above matrix zonotope is also denoted by the tuple (A^0, A^1, \dots, A^k) . The matrix A^0 is called the matrix center and the other matrices are called matrix generators.

Property 4 *Given a matrix zonotope $A^{[z]} = \{A^0 + \sum_{i=1}^k x_i A^i \mid -1 \leq x_i \leq 1, A^i \in \mathbb{R}^{n \times n}\}$ and a zonotope $Z = p \oplus HB^m$, their product is over-approximated by:*

$$\begin{aligned}
 A^{[z]}Z &= \bigcup_{x_i \in [-1,1]} \left(A^0Z \oplus \bigoplus_{i=1}^k x_i A^i Z \right) \\
 &\subseteq A^0p \oplus \left[A^0H \ A^1p \ A^1H \ A^2p \ A^2H \ \dots \ A^kp \ A^kH \right] \mathbf{B}^{m+m(k+1)}
 \end{aligned}
 \tag{48}$$

Zonotope order reduction: The proposed algorithm at each iteration includes adding zonotopes, finding the intersection of a zonotope and a strip, and multiplication of a matrix zonotope and a zonotope in the case of parametric uncertainties. Each of these operations adds to the order of the zonotope that must be stored. As a result, the order of the zonotope would increase in each iteration, which results in more computational complexity and more allocated memory. To avoid the order expansion, we limit the order of the zonotope by over-approximating it with a lower order zonotope. The following property is used for this purpose.

Property 5 [36] Given the Zonotope $Z = p + \mathbf{H}\mathbf{B}^m \subset \mathbb{R}^n$, and the integer s , where $n < s < r$, assuming that \hat{H} is the matrix resulting by reordering the columns of the matrix H in decreasing Euclidean norm, i.e.,:

$$\hat{H} = [\hat{h}_1, \dots, \hat{h}_i, \dots, \hat{h}_m] \text{ with } \|\hat{h}_i\| \geq \|\hat{h}_{i+1}\|
 \tag{49}$$

Then $Z \subseteq p \oplus \left[\hat{H}_T \ Q \right] \mathbf{B}^s$, where \hat{H}_T is a matrix that consists of the first $s - n$ columns of the matrix \hat{H} and $Q \in \mathbb{R}^{n \times n}$ is a diagonal matrix that satisfies:

$$Q_{ii} = \sum_{j=s-n+1}^m |\hat{H}_{ij}|, i = 1, \dots, n
 \tag{50}$$

6. Simulation Results

This section gives the simulation results of the proposed algorithm for fault detection of the benchmark without uncertainty. A sampling time of 0.01 s is chosen. The continuous model of the system is discretized using the zero order hold discretization method. The proposed algorithm assumes that noise is unknown but bounded, but in the Simulink model of the benchmark, sensor noises are Gaussian and hence theoretically unbounded. It is known that for a Gaussian noise, if one chooses -6σ and 6σ band, the noise will be in the band with the probability of 99.99%. Here, we use $\pm 10\sigma$ for the bound on the noise in our algorithm. The algorithm is implemented using zonotopes and the order of zonotopes is limited to 1000. First we present the simulation result using nominal parameters.

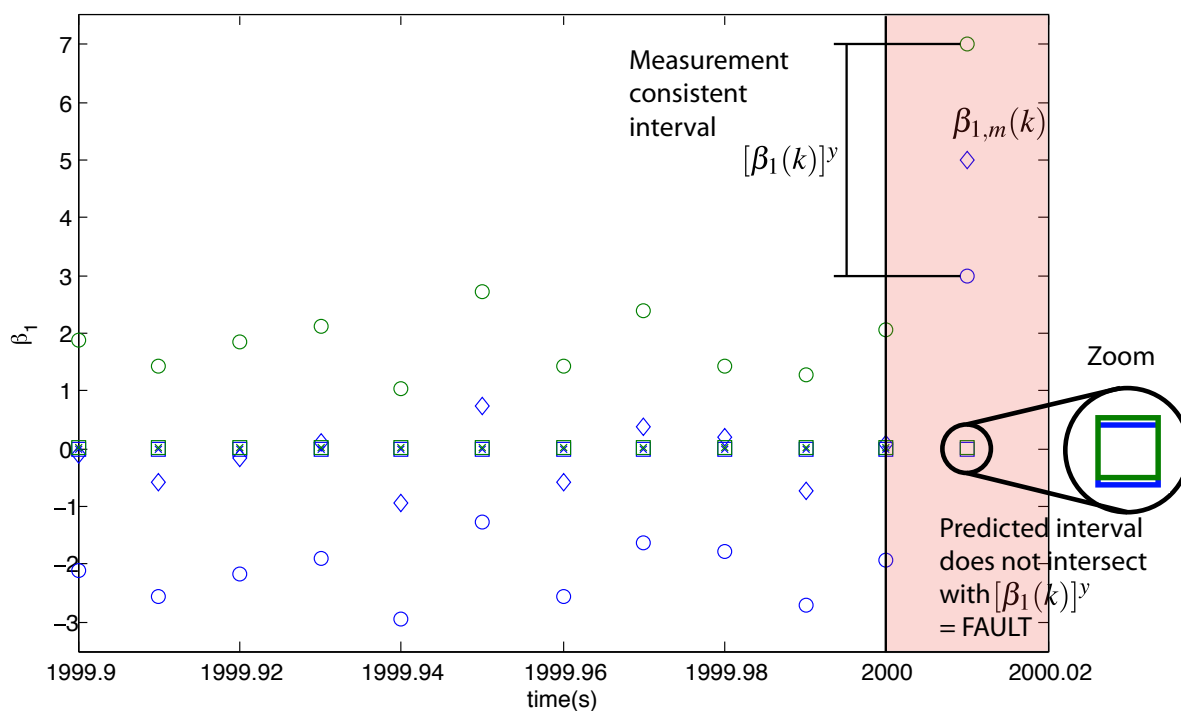
6.1. The System Without Uncertainty

6.1.1. Fault 1 ($\beta_{1,m_1} = 5^\circ$)

Figure 5 shows the simulation result for detection of fault 1. In this figure diamonds represent the measurement, circles represent the consistent interval values of β_1 with the current measurement, i.e., $[\beta_1(k)]^y$, boxes represent the interval of the predicted set, i.e., $[\beta_1(k)]^p$, and crosses represent $[\beta_1(k)]^c$. A similar notation is used in all of the following figures. The red area indicates the simulation of the faulty system until the fault is detected. As it can be seen, one sample after the occurrence of the fault,

$[\beta_1(k)]^p$ and $[\beta_1(k)]^y$ do not intersect, which means $[\beta_1(k)]^c$ is empty and the fault is detected. We have performed multiple simulations and there was no positive false alarm.

Figure 5. Detection of fault 1, \diamond : current measurement $\beta_{1,m}(k)$, \circ : $[\beta_1(k)]^y$, \square : $[\beta_1(k)]^p$, \times : $[\beta_1(k)]^c$. One sample after the occurrence of the fault, $[\beta_1(k)]^p$ and $[\beta_1(k)]^y$ do not intersect, which means $[\beta_1(k)]^c$ is empty and the fault is detected.



6.1.2. Fault 2 ($\beta_{1,m_2} = 1.2\beta_{1,m_2}$)

Due to the high level of noise on the pitch measurement, it takes some time to detect the fault. The fault is detected at 2304.21 s.

6.1.3. Fault 3 ($\beta_{3,m_1} = 10^\circ$)

This fault is detected within one sample. The result is shown in Figure 6.

6.1.4. Fault 4

This fault is a fixed value fault for the rotor speed sensor 1. Due to high level of noise on this measurement, a filtered version of this measurement is used. Fault 4 is detected in 5 samples. Figure 7 shows the result of fault detection.

Figure 6. Detection of fault 3, \diamond : current measurement $\beta_{3,m}(k)$, \circ : $[\beta_3(k)]^y$, \square : $[\beta_3(k)]^p$, \times : $[\beta_3(k)]^c$. One sample after the occurrence of the fault, $[\beta_3(k)]^p$ and $[\beta_3(k)]^y$ do not intersect, which means $[\beta_3(k)]^c$ is empty and the fault is detected.

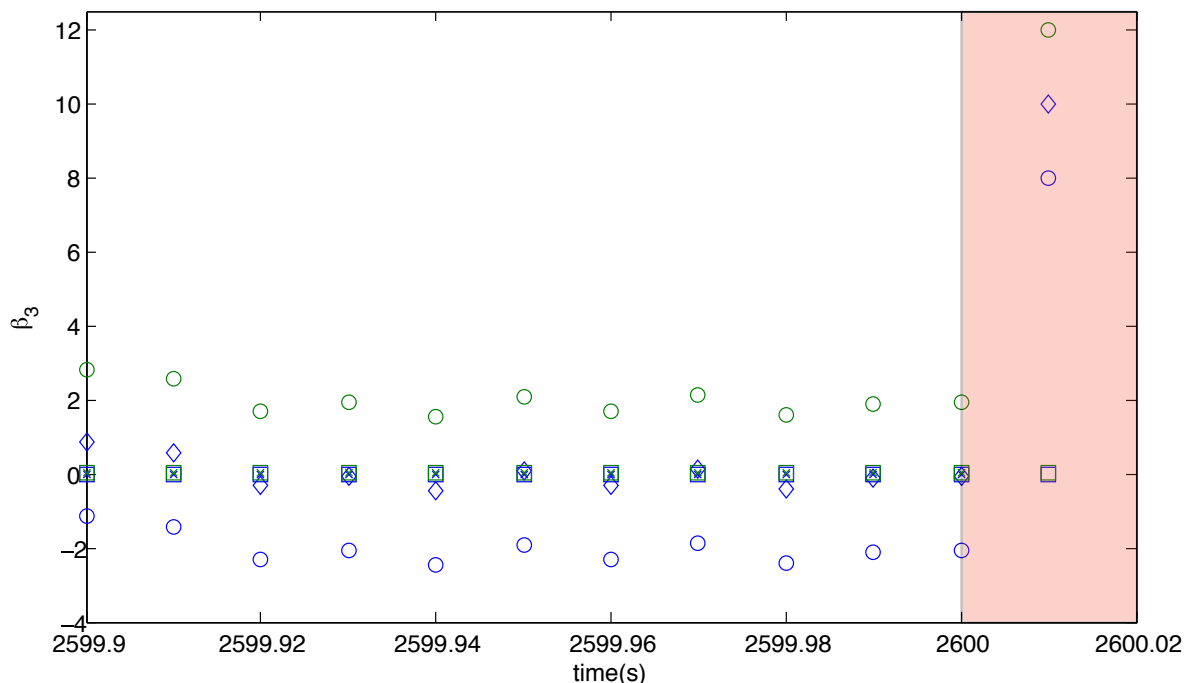
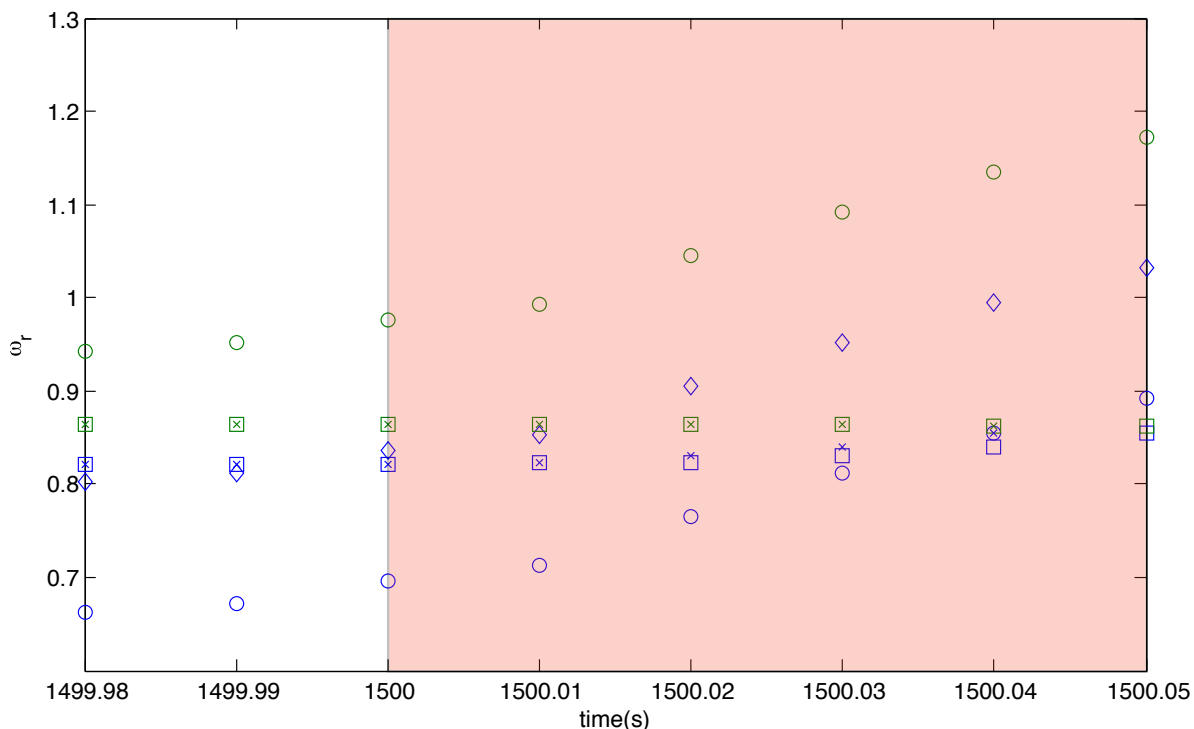


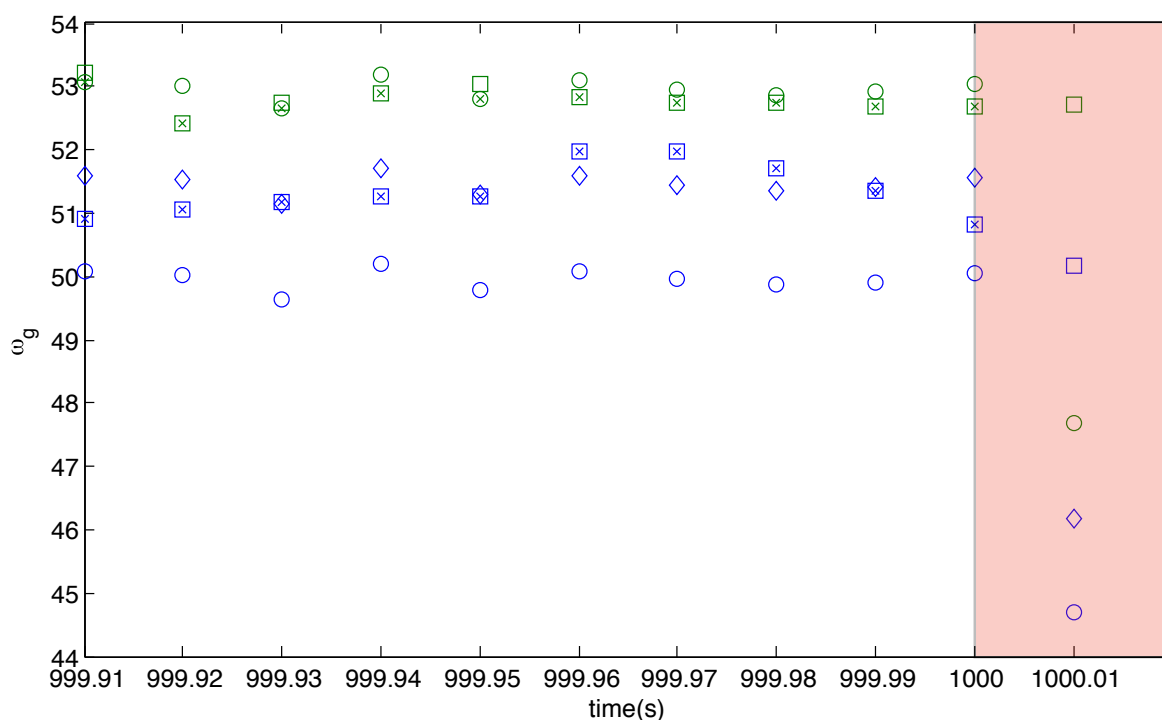
Figure 7. Detection of fault 4, \diamond : current measurement $\omega_{r,m_1}(k)$, \circ : $[\omega_{r,m_1}(k)]^y$, \square : $[\omega_{r,m_1}(k)]^p$, \times : $[\omega_{r,m_1}(k)]^c$. 5 samples after the occurrence of the fault, $[\omega_{r,m_1}(k)]^p$ and $[\omega_{r,m_1}(k)]^y$ do not intersect, which means $[\omega_{r,m_1}(k)]^c$ is empty and the fault is detected.



6.1.5. Fault 5

Fault 5 is a gain factor on the rotor speed sensor 1 and generator speed sensor 2 such that $\omega_{r,m_2} = 1.1\omega_{r,m_2}, \omega_{g,m_1} = 0.9\omega_{g,m_1}$. The fault is detected in 1 sample time. The result can be seen in Figure 8.

Figure 8. Detection of fault 5, \diamond : current measurement $\omega_{g,m_1}(k)$, \circ : $[\omega_{g,m_1}(k)]^y$, \square : $[\omega_{g,m_1}(k)]^p$, \times : $[\omega_{g,m_1}(k)]^c$. 1 sample after the occurrence of the fault, $[\omega_{g,m_1}(k)]^p$ and $[\omega_{g,m_1}(k)]^y$ do not intersect, which means $[\omega_{g,m_1}(k)]^c$ is empty and the fault is detected.



6.1.6. Fault 6

Fault 6 is hydraulic pressure drop in actuator 2. The pressure drop is modeled as abrupt change of parameters of the pitch systems from ω_n, ξ to ω_{n2}, ξ_2 . Detection of this fault is not possible until there is some big enough change in the pitch reference signal which is zero for about 50 seconds after fault occurrence. The fault is detected at 2951.69 s. This result is improved using auxiliary excitation signal as it is explained in the next subsection.

6.1.7. Fault 7

This fault is due to the air content increase in the actuator oil, which changes the parameters of the pitch actuator 3 gradually. The fault is detected at 3424.83 s choosing v_β to be $10\sigma_\beta$.

6.1.8. Fault 8

Fault 8 is a bias in the generator control loop which yields a bias in the generator torque. The fault is detected in one sample.

The results are summarized in Table 6, where T_f is the occurrence time and T_D is the detection time of a fault.

6.2. Improving Detection Time Using Auxiliary Signal Excitation

As explained, due to high level of noise and low excitation, some of the faults such as fault 6 and fault 2 are detected with a long delay. A solution to this problem is to use an auxiliary signal which is added to the original reference signal to excite the system periodically when the excitation is close to zero, such that the performance of the fault detection algorithm is increased. Here, we use an auxiliary signal to excite the pitch system by adding it to the pitch reference. For this purpose, a sinusoidal signal of the form $\beta_{aux} = 5\sin(15t) + 3$ is used. This improves the result of fault detection as summarized in Table 7. As a result of using the auxiliary excitation signal, the detection time of faults 2, 6 and 7 is decreased.

Table 6. Summary of fault detection results.

Fault No.	T_f	T_D
1	2,000 s	2,000.01 s
2	2,300 s	2,304.21 s
3	2,600 s	2,600.01 s
4	1,500 s	1,500.05 s
5	1,000 s	1,000.01 s
6	2,900 s	2,951.69 s
7	3,400 s	3,424.83 s
8	3,800 s	3,800.01 s

Table 7. Summary of fault detection results using auxiliary excitation.

Fault No.	T_f	T_D
1	2,000 s	2,000.01 s
2	2,300 s	2,300.07 s
3	2,600 s	2,600.01 s
4	1,500 s	1,500.05 s
5	1,000 s	1,000.01 s
6	2,900 s	2,900.13 s
7	3,400 s	3,400.15 s
8	3,800 s	3,800.01 s

6.3. Including Uncertainty

Two kinds of uncertainties are considered in the model, *i.e.*, uncertainties on the torque coefficient look-up table and on the system matrices. Uncertainties on the system matrices are modeled using a zonotope matrix of the form:

$$\begin{aligned} A^{[z]} &= \{A_0 + \Delta z A_0 \mid -1 \leq z \leq 1\} \\ B^{[z]} &= \{B_0 + \Delta z B_0 \mid -1 \leq z \leq 1\} \end{aligned} \quad (51)$$

where Δ denotes the level of uncertainty and A_0, B_0 are the system matrices with nominal parameters. Uncertainties on C_q are modeled by:

$$C_q = \{C_{q0} + \delta_{C_q} z C_{q0} \mid -1 \leq z \leq 1\} \quad (52)$$

Table 8 shows the result of fault detection algorithm considering 2%, 5%, and 10% system uncertainty and 5%, 10% of uncertainty on the C_q table. The symbol *ND* means that a fault is not detected. As it can be seen, when the level of uncertainty is increased, some faults such as fault No. 2 and 5 remain hidden and cannot be detected. This is because for the corresponding fault, the set of states consistent with the output of the faulty system is consistent with the prediction set based on the model of the fault-free system. Also, the detection time increases as uncertainty increases, as it is expected.

Table 8. Summary of fault detection results.

Fault No.	T_f	T_D		
		$(\Delta = 2\%, \delta_{C_q} = 5\%)$	$(\Delta = 5\%, \delta_{C_q} = 10\%)$	$(\Delta = 10\%, \delta_{C_q} = 10\%)$
1	2,000 s	2,000.01 s	2,000.01 s	2,000.01 s
2	2,300 s	<i>ND</i>	<i>ND</i>	<i>ND</i>
3	2,600 s	2,600.01 s	2,600.01 s	2,600.01 s
4	1,500 s	1,500.08 s	1,500.1 s	1,500.14 s
5	1,000 s	<i>ND</i>	<i>ND</i>	<i>ND</i>
6	2,900 s	2,900.13 s	<i>ND</i>	<i>ND</i>
7	3,400 s	3,400.22 s	<i>ND</i>	<i>ND</i>
8	3,800 s	3,800.01 s	3,800.01 s	3,800.01

6.4. Computational Complexity

Using zonotopes results in a computationally efficient method since zonotopes are efficient with respect to the set operations that are required in the algorithm. Minkowski sum of two zonotopes is simply concatenating the generators and addition of the centers. For a system with dimension n , computational complexity for the Minkowski sum of two zonotopes is $\mathcal{O}(n)$. Linear transformation of a zonotope requires linear transformation of the center and its generators. The computational complexity of linear transformation of a zonotope is $\mathcal{O}(n^3)$. For intersection, we have used the segment minimization method explained in Section 4.

To evaluate the computational complexity of the algorithm, we simulated the algorithm using Matlab R2010b on a system with 2 GB RAM, Intel[®] core i7 CPU, which runs windows 7. Since our code was not optimized for parallel processing, we set the maximum number of computational threads to be 1. Therefore, only one of the cores is used. The average processing time required for a sample is 0.0451 s. This shows that by using a faster language like C/C++, an optimized code, and a dedicated processor, it is feasible to run the code in real-time on a wind turbine. Note that another option is to decrease the sampling frequency. In the benchmark a sampling frequency of 100 Hz is used, but in a real application usually a lower sampling rate is chosen. For example, if SCADA data are used, the sampling rate is usually much lower than 100 Hz [39].

7. Conclusions

A method for fault detection of a benchmark wind turbine which includes sensor, actuator and system faults is proposed. The method is based on the state space set-membership consistency test. Instead of using wind speed measurement, we use effective wind speed estimation and then using interval analysis, the aerodynamic torque which is nonlinear is over-approximated by an interval. Using zonotopes and zonotopes matrices, we propose a computationally efficient algorithm that considers uncertainties on the system matrices and look-up table of the torque coefficient. The advantage of the method is its simplicity of tuning. Unlike residual signal based methods where designing a residual signal is a challenging task, in this method only an upper bound on the noise and uncertainty is required. Another advantage is that by using a realistic bound, no positive false alarm is produced. The simulation results demonstrate the effectiveness of the method for fault detection of the wind turbine without and with uncertainties.

Acknowledgments

The work described in this paper was supported by the Norwegian research center for offshore wind energy (NORCOWE).

References

1. Ribrant, J.; Bertling, L. Survey of failures in wind power systems with focus on Swedish wind power plants during 1997–2005. *IEEE Trans. Energy Convers.* **2007**, *22*, 167–173.
2. Esbensen, T.; Sloth, C. Fault Diagnosis and Fault-Tolerant Control of Wind Turbines. Master Thesis, Aalborg University, Aalborg, Denmark, 2009.
3. García Márquez, F.; Tobias, A.; Pinar Pérez, J.; Papaalias, M. Condition monitoring of wind turbines: Techniques and methods. *Renew. Energy* **2012**, *46*, 169–178.
4. Zhao, M.; Wang, X. State of the Art on the Monitoring Methods of Offshore Wind Energy Conversion System. *Adv. Mater. Res.* **2012**, *446*, 3100–3107.
5. Giebhardt, J. Evolutionary algorithm for optimisation of condition monitoring and fault prediction pattern classification in offshore wind turbines. In *Proceedings of European Wind Energy Conference*, Athens, Greece, 27 February–2 March 2006.
6. Caselitz, P.; Giebhardt, J. Rotor condition monitoring for improved operational safety of offshore wind energy converters. *J. Sol. Energy Eng.* **2005**, *127*, 253–261.

7. Knudsen, T.; Bak, T.; Tabatabaeipour, S.M. Detection of excessive wind turbine tower oscillations fore-aft and sideways. In *Proceedings of American Control Conference*, Montreal, Canada, 27–29 June 2012; pp. 5795–5800.
8. Ciang, C.C.; Lee, J.; Bang, H.J. Structural health monitoring for a wind turbine system: a review of damage detection methods. *Meas. Sci. Technol.* **2008**, *19*, 122001:1–122001:20.
9. Stefani, A.; Yazidi, A.; Rossi, C.; Filippetti, F.; Casadei, D.; Capolino, G. Doubly fed induction machines diagnosis based on signature analysis of rotor modulating signals. *IEEE Trans. Ind. Appl.* **2008**, *44*, 1711–1721.
10. Rothenhagen, K.; Fuchs, F. Doubly fed induction generator model-based sensor fault detection and control loop reconfiguration. *IEEE Trans. Ind. Electr.* **2009**, *56*, 4229–4238.
11. Wei, X.; Verhaegen, M.; van den Engelen, T. Sensor fault diagnosis of wind turbines for fault tolerant. In *Proceedings of the 17th International Federation of Automatic Control World Congress (IFAC)*, Seoul, Korea, 6–11 July 2008; pp. 3222–3227.
12. Odgaard, P.; Stoustrup, J. Unknown input observer based detection of sensor faults in a wind turbine. In *Proceeding of IEEE International Conference on Control Applications*, Yokohama, Japan, 8–10 September 2010; pp. 310–315.
13. Poure, P.; Weber, P.; Theilliol, D.; Saadate, S. Fault-tolerant power electronic converters: Reliability analysis of active power filter. In *Proceedings of IEEE International Symposium on Industrial Electronics*, Vigo, Spain, 4–7 June, 2007; pp. 3174–3179.
14. Odgaard, P.; Stoustrup, J.; Kinnaert, M. Fault tolerant control of wind turbines a benchmark model. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Barcelona, Spain, 30 June–3 July 2009; pp. 155–160.
15. Chen, W.; Ding, S.; Sari, A.; Naik, A.; Khan, A.; Yin, S. Observer-based FDI schemes for wind turbine benchmark. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 7073–7078.
16. Laouti, N.; Sheibat-Othman, N.; Othman, S. Support vector machines for fault detection in wind turbines. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 7067–7072.
17. Ozdemir, A.; Seiler, P.; Balas, G. Wind turbine fault detection using counter-based residual thresholding. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 8289–8294.
18. Svärd, C.; Nyberg, M. Automated Design of an FDI System for the Wind Turbine Benchmark. *J. Cont. Sci. Eng.* **2011**, *2012*, 1–13.
19. Zhang, X.; Zhang, Q.; Zhao, S.; Ferrari, R.M.; Polycarpou, M.M.; Parisini, T. Fault detection and isolation of the wind turbine benchmark: An estimation-based approach. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 8295–8300.
20. Stoican, F.; Raduinea, C.F.; Olaru, S. Adaptation of set theoretic methods to the fault detection of wind turbine benchmark. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 8322–8327.

21. Kiasi, F.; Prakash, J.; Shah, S.; Lee, J. Fault detection and isolation of benchmark wind turbine using the likelihood ratio test. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 7079–7085.
22. Simani, S.; Castaldi, P.; Bonfe, M. Hybrid model-based fault detection of wind turbine sensors. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 7061–7066.
23. Simani, S.; Castaldi, P.; Tilli, A. Data-driven approach for wind turbine actuator and sensor fault detection and isolation. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 8301–8306.
24. Ayalew, B.; Pisu, P. Robust fault diagnosis for a horizontal axis wind turbine. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 7055–7060.
25. Chen, J.; Patton, R. *Robust Model-Based Fault Diagnosis for Dynamic Systems*; Kluwer Academic Publishers: Norwell, MA, USA, 1999.
26. Patton, R.; Chen, J. Robust fault detection of jet engine sensor systems using eigenstructure assignment. In *Proceedings of AIAA Guidance, Navigation and Control Conference*, New Orleans, LA, USA, 1991; pp. 1666–1675.
27. Patton, R.; Chen, J. A review of parity space approaches to fault diagnosis. In *Proceedings of IFAC SAFEPROCESS Symposium*, Baden-Baden, Germany, 10–13 September 1991; pp. 65–81.
28. Puig, V. Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies. *Int. J. Appl. Math. Comput. Sci.* **2010**, *20*, 619–635.
29. Ingimundarson, A.; Bravo, J.M.; Puig, V.; Alamo, T.; Guerra, P. Robust fault detection using zonotope-based set-membership consistency test. *Int. J. Adapt. Control Signal Process.* **2009**, *23*, 311–330.
30. Blesa, J.; Puig, V.; Romera, J.; Saludes, J. Fault diagnosis of wind turbines using a set-membership approach. In *Proceedings of the 18th International Federation of Automatic Control World Congress (IFAC)*, Milan, Italy, 28 August–2 September 2011; pp. 8316–8321.
31. Bianchi, F.; De Battista, H.; Mantz, R. *Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Design*; Springer Verlag: London, UK, 2007.
32. Johnson, K.; Pao, L.; Balas, M.; Fingersh, L. Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture. *IEEE Control Syst. Mag.* **2006**, *26*, 70–81.
33. Merritt, H. *Hydraulic Control Systems*; John Wiley & Sons Inc.: New York, NY, USA, 1967.
34. Østergaard, K.; Brath, P.; Stoustrup, J. Estimation of effective wind speed. *J. Phys. Conf. Ser.* **2007**, *75*, 012082:1–012082:9.
35. Knudsen, T.; Bak, T.; Soltani, M. Prediction models for wind speed at turbine locations in a wind farm. *Wind Energy* **2011**, *14*, 877–894.
36. Alamo, T.; Bravo, J.; Camacho, E. Guaranteed state estimation by zonotopes. *Automatica* **2005**, *41*, 1035–1043.
37. Vicino, A.; Zappa, G. Sequential approximation of feasible parameter sets for identification with set membership uncertainty. *Autom. Control IEEE Trans.* **1996**, *41*, 774–785.

38. Althoff, M.; Stursberg, O.; Buss, M. Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear Anal. Hybrid Sys.* **2010**, *4*, 233–249.
39. Kim, K.; Parthasarathy, G.; Uluoyol, O.; Foslien, W.; Sheng, S.; Fleming, P. *Use of SCADA Data for Failure Detection in Wind Turbines*; Technical Report; National Renewable Energy Laboratory (NREL), Golden, CO, USA, 2011.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).