Departamento de Ingeniería de Sistemas y Automática

Máster en Automática e Informática Industrial

FAULT DIAGNOSIS OF WIND TURBINES APPLICATION TO A BENCHMARK MODEL

Tesina de Máster

Autor: Héctor Eloy Sánchez Sardi

Director: Dr. Emilio García Moreno

Codirectora: Dra. Teresa Escobet Canal

Codirector: Dr. Vicenç Puig Cayuela

Septiembre 2013, Valencia

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today wind turbines contribute to a larger and larger part of the world's power production, at the same time the size of the standard turbine is increasing. The US targets 20% wind-based electricity generation, i.e. over 300 GW, by 2030 [2].

The TPWind envisions the coverage of European Union electricity generation up to 12-14% by 2020 and 25% of by 2030 [39]. China aims for 15% renewable power generation by 2020 [1].

Wind turbines in the megawatt size as most often installed at present, are expensive. A major issue with wind turbines systems specially those located offshore, is the relatively high cost of operation and maintenance (OM). Wind turbines are hard-to-access structures, and they are often located in remote areas. These factors alone increase the OM cost for wind power systems. Also, poor reliability directly reduces availability of wind power due to the turbine downtime [45].

According to General Electric (GE) Energy, a $5,000 bearing replacement can easily turn into a $250,000 project involving cranes, service crew, gearbox replacements, and generator rewinds, not to mention the downtime loss of power generation [15]. For a turbine with over 20 years of operating life, the OM and part costs are estimated to be 10-15% of the total income for a wind farm [45]. Although larger turbines may reduce the OM cost per unit power, the cost per failure is increased. The OM cost for offshore wind turbine is estimated to be 20-25% of the total income [26], [29]. There are research interests in developing automatic maintenance systems for offshore wind turbines which aim to reduce the costs detailed before, see for example [27].

Condition monitoring and fault diagnosis of wind turbines has thus greater benefit for such situations. In addition, wind turbine repair and maintenance that require extensive usage of cranes and lifting equipment create a highly capital-intensive operation as well as delayed services due to lack of crane availability and needs for optimal weather conditions. Also, the trend that has currently emerged to dampen prospects is lack of personnel available to perform the consistent OM required to keep turbines functioning and efficient.

There have been a few literature reviews on wind turbine condition monitoring in literatures [50] [49] [38] [25] . However, as the renewable energies have gained dramatically increasing attention from industries and academia since 2006, many new research works have been reported in the condition monitoring and fault diagnosis areas.

Therefore the reliability of these turbines is important. Their off time should be as little as possible. An important part of ensuring this is to introduce advanced fault detection, isolation and accommodation systems on the wind turbines. In the state-of-the-art industrial wind turbines, fault detection and accommodation schemes are simple and are often conservative. Consequently, the use of advanced fault detection, isolation and accommodation methods could improve the reliability of the turbine, even though; it might result in production with limited power for some faults.

Alternatively autonomous online condition monitoring systems with integrated fault detection algorithms allow early warnings of mechanical and electrical faults to prevent major component failures.

Side effects on other components such as gear box, bearings, blades pitch, rotor, generator, the braking system and several system sensors can be reduced significantly with online condition monitoring and diagnosis. Many faults can be detected while the defective component is still operational. Therefore, necessary repair actions can be planned in time and need not to be taken immediately. This is important especially

for offshore plants, where bad weather conditions (storms, high tides, etc) can prevent any repair actions for several weeks. Also, condition monitoring and fault diagnosis can detect extreme external conditions, such as icing or water induced tower oscillations of offshore plants, and can trigger appropriate control actions to prevent damage of wind turbine components.

Early fault detection and diagnosis allow the operators in an industrial process to take the best actions during the real state of the process, avoiding incipient faults to scale to critical situations where there is risk of human lives and economical lost [4], this is specially true in offshore wind turbines. As a consequence, recently has been emerged the interest in prognosis and predictive maintenance, to be able to predict the remaining useful life of a machine and to schedule maintenance prior to the failure of a component will reduce the time of a wind turbine shutdown and even the operation of the wind turbine itself.

The work done in this master thesis goes in the line of the CICYT project "System Health Management and Reliable Control of Complex Systems" DPI2011-26243. The main goal of this project is to develop a set of advanced tools for the reliable control and health management of complex systems. These tools will provide better availability, dependability and safety conditions of industrial processes and/or systems, therefore improving their overall performance. To obtain these new tools, new methodologies must be developed, through the integration of fault diagnosis, fault tolerant techniques and predictive maintenance. Reliable control techniques will increase the system safety and availability, once fault detection and diagnosis are performed. Moreover, predictive maintenance will increase the availability and dependability of the system, preventing fault appearance through a precise estimation of the remaining usable life-time in system components/processes.

This master thesis is aimed in part to achieve one of the workpackages of the project explained before: Development of fault diagnosis algorithms for health monitoring that take into account information from multiple sources (signal analysis, or model-based estimations; time or frequency domains) and well cover the full cycle of system monitoring and supervision.

In this chapter is given a brief overview of the state of the art in fault diagnosis of wind turbines, it is explained the motivation and objectives of this thesis and the structure of this document is resumed.

1.1   State of the Art

In the state-of-the-art industrial wind turbines fault detection and accommodation schemes are simple and are most often conservative. Consequently the use of advanced fault detection, isolation and accommodation methods could improve the reliability of the turbine, even though, for some faults, it might result in production with limited power. Alternatively condition monitoring is used to monitoring some mechanical components such as gear boxes, etc, see [50].

Some work has been performed on model based fault detection, isolation and accommodation on wind turbines. In [47], an observer based scheme to detect sensor faults in the pitch system was presented. A parity equations based scheme for fault detection on wind turbines was presented in [8], an unknown input observer was proposed for detection of sensor faults around the wind turbine drive train in [35]. Fault detection of electrical conversion systems can be found as .f.e in [36].

Modern wind turbine control systems are equipped with condition monitoring and fault detection systems. These systems detect and isolate faults and determine the current operating conditions of the wind turbine. The available information can then be utilized for predictive maintenance, which basically predicts when maintenance should be performed to avoid failures.

Most condition monitoring and fault detection systems in wind turbines are signal-based and utilize e.g. vibration analysis to detect and isolate faults. This has enabled successful condition monitoring of bearings in the gearbox and the generator among others. Only a few model-based fault diagnosis approaches exist for wind turbines; among these are fault diagnosis systems for pitch sensors and actuators [46] and [9]. These diagnosis systems estimate some parameters in the pitch system, and determine if a fault has occurred based on these estimates.

In [11], an structural analysis of a wind turbine system was done and some analytical redundant relations were found to perform fault detection. Their fault diagnosis system consists of fault detection and isolation algorithms that determine the current state of the system and reconfigure an extended Kalman filter, which is able to provide a fault-corrected state estimate at all times. This last work was taken as an important

reference in the development of this master thesis.

In [5], a fault diagnosis system for a wind turbine was designed using model-based fault diagnosis methods, real field data from a 3MW wind turbine was used in this thesis to identify a nominal model and Linear Parameter Varying (LPV) models were used to do fault diagnosis in a wide range of operating points.

In [37], a benchmark model of a wind turbine at system level, containing: sensors, actuators and systems faults was presented. This benchmark model was based on a realistic generic three blade horizontal variable speed wind turbine with a full converter coupling. In the spring of 2010 kk-electronic together with other partners (MathWorks and Aalborg University) launched an international competition on Fault Detection and Isolation in Wind Turbines based on this benchmark model of the wind turbine. The competition consisted to find the best schemes to detect and handle the different faults proposed, and the results of the competition were presented in [33]. The competition drew enough high-quality papers to fill two sessions at the IFAC World Congress 2011 and also several proposals were published at the IFAC Safeprocess in 2012.

Some of the solutions commented in: [33] were the following:

- This solution uses a diagnostic observer based residual generator for the residual generation [44].

- The solution proposed in [31] used a Support Vector Machine based on a Gaussian kernel.

- In [3] a counters solution was used for decision of fault detection and isolation based on residuals generated using physical and analytical redundancy.

- In [40] it was used a general and automatic fault detection solution.

- A fault detection estimator was used in [48] to estimate the presence of a fault.

- In [22] a model based fault diagnosis approach called set-membership approach was used.

After the announcement of results of the first benchmark, a second challenge was presented in [32], this new benchmark differed from the previous challenge in several ways. The second benchmark wind turbine model is modeled in FAST simulator [20]. In this case there are no sensor models available and the whole wind turbine dynamics are implemented in the simulator differing from the first benchmark in which all the subsystem models were provided.

1.2 Motivation and Thesis Objectives

The new proposed benchmark requires sophisticated fault detection and fault-tolerant control tools and likely making the results of greater applicability to the wind industry. This higher-fidelity model also allows the use of more realistic wind inputs that vary spatially across the rotor plane in addition to temporally. Also, the fault scenarios have been updated and additional information detailing their relevance has been provided.

The new challenge proposed on [32] differs from the initial benchmark [37] in the fact that many of the wind turbines subsystems models are not available, making necessary to find new models in the literature and calibrate these models in order apply model based fault diagnosis.

The characteristics of the new benchmark providing a higher fidelity and more realistic model, puts a motivation to find and adjust suitable models that represent in the best way possible the dynamics of the wind turbine system modeled in this second benchmark. This is pursued in order to perform a more reliable model based fault diagnosis.

The fault scenarios outlined in the second challenge, comes from the research and experience of very recognized researchers in academia and the industry. To develop a fault diagnosis system that can detect and isolate the faults proposed in the benchmark and studied in this thesis, which are a list of common faults found in wind turbines today, becomes more feasible the possibility to extrapolate the results to a real wind turbine.

After the introduction and motivation explained before, the main objective intended with this Master Thesis is to implement a fault diagnosis algorithm on this realistic benchmark that can improve the reliability of a wind turbine.

To reach the main objective stated in the previous paragraph, the following tasks should be carried out:

- Review the State of the Art of Fault Diagnosis applied to Wind Turbines.

- Describe the most common faults in Wind Turbines.

- Study the benchmark model and the specifications related to fault diagnosis.

- Explore the techniques that can be used in the benchmark case of study.

- Select a Fault Diagnosis Technique and apply it to the case of study.

- Analysis of Results and extrapolation to a real case.

1.3   Thesis Outline

The structure of the document is divided in chapters and resumed in this section to provide an overview of the thesis.

Chapter 2: Wind Turbine Modeling. This chapter provides a description of the wind turbine system and a model for each one of the subsystems that form the wind turbine are proposed. Different subsystem models from the literature are examined and compared between them to proceed with the selection of the model to be used in the next stage which is the model based fault diagnosis.

Chapter 3: Structural Analysis. The purpose of this chapter is to perform an structural analysis of the wind turbine system model proposed in chapter 3 in order to determine the analytical redundant relations which are used to detect and isolate the faults in the fault diagnosis system implemented and explained in chapter 6.

Chapter 4: Calibration, Parameters Adjustment and System Identification. The motivation of this chapter is to adjust and parameterize the theoretical models and analytical redundant relations obtained in chapter 4 based on the behavior observed in FAST measured wind turbine variables. This is done in order to have models that approximate in the best way possible the dynamics observed in FAST simulator to pursue a good and reliable performance of the fault diagnosis system implemented in chapter 6.

Chapter 5: Fault Diagnosis System Implementation. This chapter illustrates the fault diagnosis system implementation in which all the obtained residuals are implemented and tested with the different fault scenarios proposed in the benchmark [32].

Chapter 6: Conclusions. The conclusions of this thesis, as well the limitations and future work is commented in this chapter.

Chapter 2

Wind Turbine Description and Modeling

The purpose of this chapter is to give the general description of the wind turbine and set up a mathematical model of the considered wind turbine. The model should be detailed enough in order to understand the system behavior.

2.1   System Description

This section briefly describes the components of the wind turbine considered in this work, and it was taken as a reference the work of [41]. The wind turbine components are showed in Figure 2.1.

The figure is taken as reference [16]. The components and their purposes are described below in alphabetic order [34].

- **Anemometer** is used to measure the wind speed. The wind turbine is started when the wind speed reaches a lower limit, while operation is cut-out when wind speeds become too high.

- **Brakes** can be applied mechanically, electrically, or hydraulically and function as parking brakes.

- **Gearbox** connects the low-speed shaft to the high-speed shaft, thus increasing the rotational speed to a level required by the generator to produce electric energy.

- **Generator** converts rotational energy into electric energy. On a modern wind turbine the maximum power output is typically from a few and up to 7.6 megawatts [18].

- **High-speed shaft** drives the generator.

- **Hub and rotor blades** together make up the rotor of the wind turbine. The hub connects the rotor blades to the low-speed shaft. Pitching the blades is used to maximize the efficiency in low winds and reduce efficiency in high winds to protect the wind turbine from structural damage.

- **Low-speed shaft** connects the rotor to the gearbox.

- **Nacelle** is located atop the tower and contains the gearbox, low- and high-speed shafts, generator, and brakes.

- **Tower** carries the nacelle and the rotor. Since the wind speed increases with the height, a taller tower generally enables a wind turbine to generate more electric energy.

- **Wind vane** is used to measure the direction of the wind. The wind direction is used by the yaw mechanism to orient the wind turbine perpendicular to the wind.

- **Yaw mechanism** uses electrical motors to orient the wind turbine rotor perpendicular to the direction of the wind.
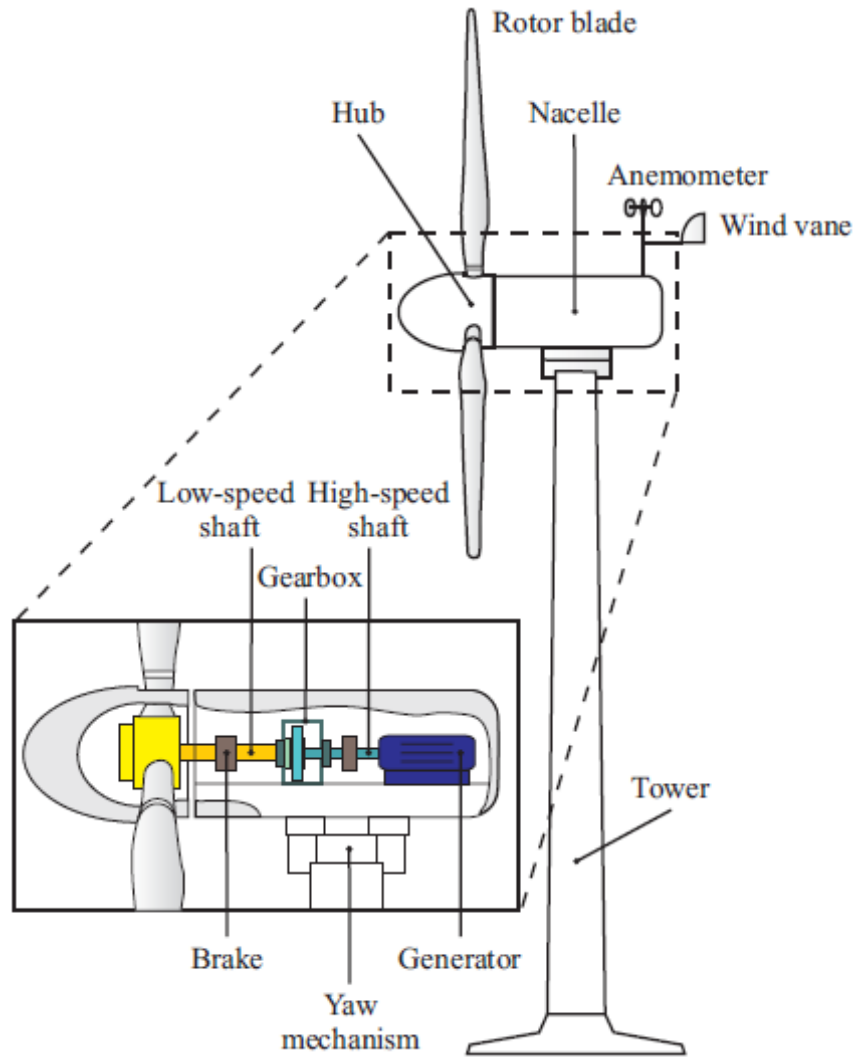
Figure 2.1: Main components of a horizontal-axis wind turbine.

2.2   Wind Turbine Model

In the following section the structure of the wind turbine model is presented in a block diagram. Afterwards, each sub-model of the wind turbine is presented and combined to obtain a complete model of the wind turbine. The overall wind turbine system model is divided into appropriate sub-models suitable of being modeled separately. Linear models for each subsystem were considered and the interaction between these subsystems results in the operation of the complete wind turbine system as illustrated in Figure Figure 2.2.

The wind speed $v_w(t)$ is the driving force of the system. As the wind blows over the turbine's blades they create "lift", much like an airplane wing, and begin to turn. Most turbines have three large blades that are aerodynamically designed to turn as easily as possible when the wind blows on them. These turning blades spin a shaft normally to some 30 to 60 times every minute (low-speed shaft). The gearbox connects the low-speed shaft with a high speed shaft that drives the generator. The gears also boost the rotation speed of the high-speed shaft to 1000 to 1800 rotations per minute. This rapidly spinning shaft drives the generator to produce electric power. The generator's electrical output is connected to a electrical grid. This general explanation of the wind turbine operation is referenced from by [17].

The aerodynamic properties of the wind turbine are affected by the pitch angles of the blades, the speed
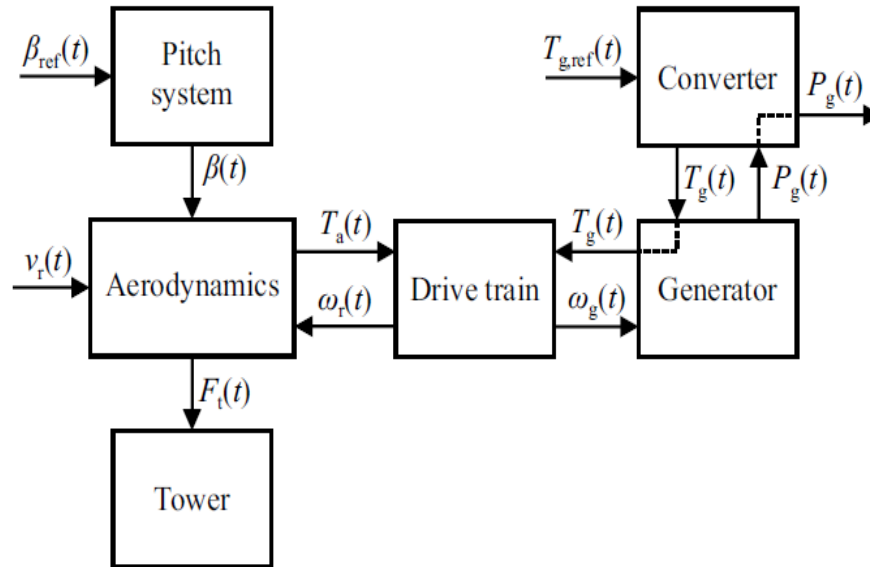
Figure 2.2: Subsystems models interaction of the wind turbine system model.

of the rotor, and the wind speed. On this basis, an aerodynamic torque is transferred from the rotor to the generator through the drive train, and an aerodynamic thrust affects the rotor and thereby the tower. The output of the wind turbine is electric power which comes from the converter. To operate the wind turbine according to the set of operating requirements, the pitch angles of the blades and the generator torque are adjusted. A pitch system controls the pitch angles of the blades, while a converter controls the generator torque. In this section, the wind turbine model has been divided into six sub-models, in order to be individually modeled and combined afterwards.

2.3 Drive Train Subsystem

The function of the drive train is to step up the speed of the low-speed shaft (rotor) to a suitable value for the generator in order to produce electrical power.

As mentioned in [42], in general, a drive train model consists of the following elements connected in series:

- A body with rotational inertia and damping (representing the turbine rotor).
- A torsional spring (representing the gearbox).
- A body with rotational inertia (representing the generator rotor).
- A torsional damper (modeling the resistance produced by slip on the induction generator). Based on the physical laws, the authors from the consulted literature proposed different drive train models and three of them are analyzed in this section. These three models come from the main references encountered. The first one of them is developed in [13], and consists in a single shaft drive train. The second one is proposed by [11] and consists in two shafts linked together by a gearbox with its respective gear ratio. The third one is the drive train model proposed in the benchmark [37].

The drive train models can be obtained using Lagrange's energy equations or Newton's equations as will be showed in the next subsections.

At the end of the section after reviewing the three models a drive train model is proposed to carry on further fault diagnosis analysis.

2.3.1 Single Shaft Model

In [13], the drive train subsystem is modeled as two rigid bodies linked by a flexible shaft, as is showed in Figure 2.3.

The drive train is modeled as two rigid bodies linked by a flexible shaft. The rigid bodies encompass all the mechanical devices and parts of them located at each side of the effective shaft. Accordingly, the terms moment of inertia of the rotor (J_r), moment of inertia of the generator (J_g), torsion stiffness of the drive train (K_{dt}) and torsion damping coefficient of the drive train (B_{dt}) denote model parameters, rather than physical ones.
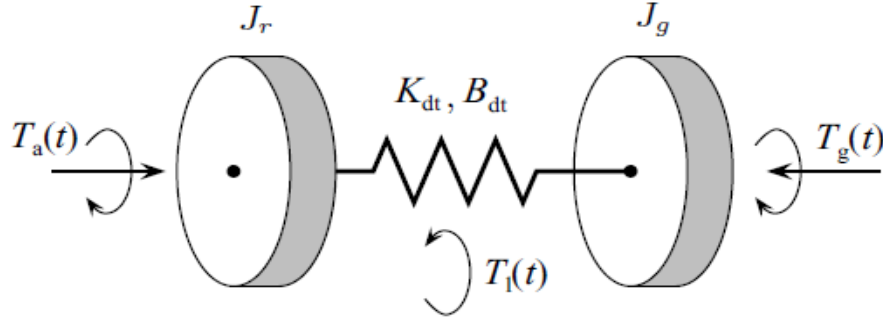
Figure 2.3: Drive-train

A mechanical system of arbitrary complexity can be described by the equation of motion

$$M\ddot{q} + C\dot{q} + Kq = Q(\dot{q}, q, t, u) \tag{2.1}$$

where M, C and K are the mass, damping and stiffness matrices and Q is the vector of forces acting on the system. For mechanical structures having few degrees of freedom, the Lagrange's equation

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{q}_i}\right) - \frac{\partial E_k}{\partial q_i} + \frac{\partial E_d}{\partial \dot{q}_i} + \frac{\partial E_p}{\partial q_i} = Q_i \tag{2.2}$$

offers a systematic procedure to derive mathematical models. E_k, E_d and E_p denote the kinetic, dissipated and potential energy, respectively. Besides, q_i is the generalized coordinate and Q_i stands for the generalized force.

For the model of equation (2.2), has been adopted the following generalized coordinates:

$$q = [\theta_r \; \theta_g]^T, \tag{2.3}$$

where: $\theta_r = q_1$ and $\theta_g = q_2$ are the angles of the rotor and generator, respectively. After these definitions, the energy terms E_k, E_d and E_p can be written as:

$$E_k = \frac{J_r}{2}\omega_r^2 + \frac{J_g}{2}\omega_g^2, \tag{2.4}$$

$$E_d = \frac{B_{dt}}{2}(\omega_r - \omega_g)^2, \tag{2.5}$$

$$E_p = \frac{K_{dt}}{2}(\theta_r - \theta_g)^2, \tag{2.6}$$

where ω_r and ω_g are the rotational speeds of the rotor and generator, respectively, both of them referred to the low-speed side of the wind turbine. The remaining parameters are defined in Table 2.1.

The vector of generalized loads is:

$$Q = [T_a \; -T_g]^T. \tag{2.7}$$

where $T_a(t)$ is the aerodynamic torque applied to the rotor and $T_g(t)$ is the generator torque.

Then, replacing the equations in the Lagrange's equation yields the motion equation(2.1).

Applying the Lagrange's equation (2.2) to each of the the corresponding energy terms (2.4)-(2.7), the following equations are obtained:

Table 2.1: Parameters of the drive-train subsystem

Parameter	Units	Description
J_r	kg m^2	Moment of inertia of the rotor
J_g	kg m^2	Moment of inertia of the generator
K_{dt}	N m/rad	Torsion stiffness of the drive train
B_{dt}	N m/(rad/s)	Torsion damping coefficient of the drive train

For the first generalized coordinate $\theta_r = q_1$ the derivative is denoted as $\dot{q}_1 = \omega_r$. The Lagrange terms for q_1 are developed below.

$$\frac{\partial E_k}{\partial \dot{q}_1} = \frac{\partial E_k}{\partial \omega_r} = J_r \omega_r \tag{2.8}$$

$$\frac{\partial E_k}{\partial q_1} = 0, \tag{2.9}$$

$$\frac{\partial E_d}{\partial \dot{q}_1} = \frac{\partial E_d}{\partial \omega_r} = \frac{B_{dt}}{2} \left(2\omega_r - 2\omega_g\right) = B_{dt} \left(\omega_r - \omega_g\right), \tag{2.10}$$

$$\frac{\partial E_p}{\partial q_1} = \frac{\partial E_p}{\partial \theta_r} = \frac{1}{2} K_{dt} \left(2\theta_r - 2\theta_g\right) = K_{dt} \left(\theta_r - \theta_g\right), \tag{2.11}$$

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \omega_r}\right) = J_r \dot{\omega}_r, \tag{2.12}$$

$$Q_1 = T_a. \tag{2.13}$$

Substituting the equations (2.8)-(2.13) in the Lagrange equation (2.2), the dynamic of the rotor is:

$$J_r \dot{\omega}_r + B_{dt} \left(\omega_r - \omega_g\right) + K_{dt} \left(\theta_r - \theta_g\right) = T_a(t) \tag{2.14}$$

For the second generalized coordinate $\theta_g = q_g$, the derivative is denoted as $\dot{q}_2 = \omega_g$. The Lagrange terms for q_2 are as follows.

$$\frac{\partial E_k}{\partial \dot{q}_2} = \frac{\partial E_k}{\partial \omega_g} = J_g \omega_g \tag{2.15}$$

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \omega_g}\right) = J_g \dot{\omega}_g, \tag{2.16}$$

$$\frac{\partial E_k}{\partial \theta_g} = 0, \tag{2.17}$$

$$\frac{\partial E_d}{\partial \omega_g} = \frac{B_{dt}}{2} \left(2\omega_g - 2\omega_r\right) = B_{dt} \left(\omega_g - \omega_r\right), \tag{2.18}$$

$$\frac{\partial E_p}{\partial q_2} = \frac{\partial E_p}{\partial \theta_g} = \frac{1}{2} K_{dt} \left(-2\theta_r + 2\theta_g\right) = K_{dt} \left(\theta_g - \theta_r\right), \tag{2.19}$$

$$Q_2 = -T_g. \tag{2.20}$$

Substituting the equations (2.15)-(2.20) in the Lagrange's equation (2.2), the dynamic of the generator is:

$$J_g \dot{\omega}_g + B_{dt} \left(\omega_r - \omega_g\right) - K_{dt} \left(\theta_r - \theta_g\right) = -T_g(t) \tag{2.21}$$

In order to reduce the formulas complexity, the absolute angular positions of the shafts θ_r and θ_g were replaced with a single state variable $\theta_\Delta = \theta_r - \theta_g$ denoting the torsion angle of the drive train.

After this substitution, the state model of the drive can be written as:

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx, \end{cases} \tag{2.22}$$

where the state, input and output vectors are

$$x = [\omega_r \ \ \omega_g \ \ \theta_\Delta]^T, \tag{2.23}$$

$$u = [T_a \ \ T_g]^T, \tag{2.24}$$

$$y = [\omega_r \ \ \omega_g]^T, \tag{2.25}$$

and the matrices A, B and C are:

$$A = \begin{bmatrix} -\frac{B_{dt}}{J_r} & \frac{B_{dt}}{J_r} & -\frac{K_{dt}}{J_r} \\ \frac{B_{dt}}{J_g} & -\frac{B_{dt}}{J_g} & \frac{K_{dt}}{J_g} \\ 1 & -1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} \frac{1}{J_r} & 0 \\ 0 & -\frac{1}{J_g} \\ 0 & 0 \end{bmatrix}, \tag{2.26}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The same model can be obtained using the Newton's equations as expressed below.

For the rotor the dynamics are:

$$J_r \ddot{\theta}_r(t) = T_a(t) - T_1(t) \tag{2.27}$$

where the torsion of the drive train is modeled using a torsion spring and a friction coefficient model according to:

$$T_1(t) = K_{dt}\theta_\Delta(t) + B_{dt}\dot{\theta}_\Delta(t) \tag{2.28}$$

Substituting (2.28) in (2.27) the following expressions are obtained.

$$J_r \ddot{\theta}_r(t) = T_a(t) - K_{dt}\theta_\Delta(t) - B_{dt}\dot{\theta}_\Delta(t) \tag{2.29}$$

$$\ddot{\theta}_r(t) = \frac{1}{J_r}T_a(t) - \frac{1}{J_r}K_{dt}\theta_\Delta(t) - \frac{1}{J_r}B_{dt}\dot{\theta}_\Delta(t) \tag{2.30}$$

The torsion angle is defined as:

$$\theta_\Delta(t) = \theta_r(t) - \theta_g(t) \tag{2.31}$$

The following change to denote the angular speed of the rotor and generator is made

$$\dot{\theta}_r = \omega_r \\ \dot{\theta}_g = \omega_g \tag{2.32}$$

The dynamic of the rotor is defined as:

$$\dot{\omega}_r(t) = \frac{1}{J_r}T_a(t) - \frac{1}{J_r}K_{dt}\theta_\Delta(t) - \frac{1}{J_r}B_{dt}(\omega_r - \omega_g) \tag{2.33}$$

The same method is used to obtain the dynamics of the generator

$$J_g \ddot{\theta}_g(t) = T_l(t) - T_g(t) \tag{2.34}$$

The torsion of the drive train is the same for the rotor and the generator, therefore:

$$T_l(t) = K_{dt}\theta_\Delta(t) + B_{dt}\dot{\theta}_\Delta(t) \tag{2.35}$$

Substituting (2.35) in (2.34) the following equation is obtained

$$J_g\dot{\omega}_g(t) = K_{dt}\theta_\Delta(t) + B_{dt}\dot{\theta}_\Delta(t) - T_g(t) \tag{2.36}$$

The dynamics for the generator is the following:

$$\dot{\omega}_g(t) = \frac{K_{dt}}{J_g}\theta_\Delta(t) + \frac{B_{dt}}{J_g}(\omega_r - \omega_g) - \frac{T_g(t)}{J_g} \tag{2.37}$$

The state space of the drive-train reads:

$$\dot{x} = \begin{bmatrix} -\frac{B_{dt}}{J_r} & \frac{B_{dt}}{J_r} & -\frac{K_{dt}}{J_r} \\ \frac{B_{dt}}{J_g} & -\frac{B_{dt}}{J_g} & \frac{K_{dt}}{J_g} \\ 1 & -1 & 0 \end{bmatrix} x + \begin{bmatrix} \frac{1}{J_r} & 0 \\ 0 & -\frac{1}{J_g} \\ 0 & 0 \end{bmatrix} u \tag{2.38}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x \tag{2.39}$$

where:

$$\begin{aligned} x &= [\omega_r \ \omega_g \ \theta_\Delta]^T \\ u &= [T_a \ T_g]^T \\ y &= [\omega_r \ \omega_g]^T \end{aligned} \tag{2.40}$$

As seen in this section both methodologies, Lagrange and Newton allow to obtain the same state space models.

2.3.2 Two Shafts Model

A two shafts model consisting of a high speed shaft and a low speed shaft linked by a gearbox is proposed by [11]. The aerodynamic torque is transferred to the generator through the drive train in order to upscale the rotational speed of the rotor, to a higher speed required by the generator.

The drive train model includes a low-speed shaft and a high-speed shaft, each composed of a moment of inertia and a frictional coefficient as illustrated in Figure 2.4. The shafts are linked together by a gearbox modeled as a gear ratio without any loss. To describe the flexibility of the drive train, a torsion spring is included in the model.
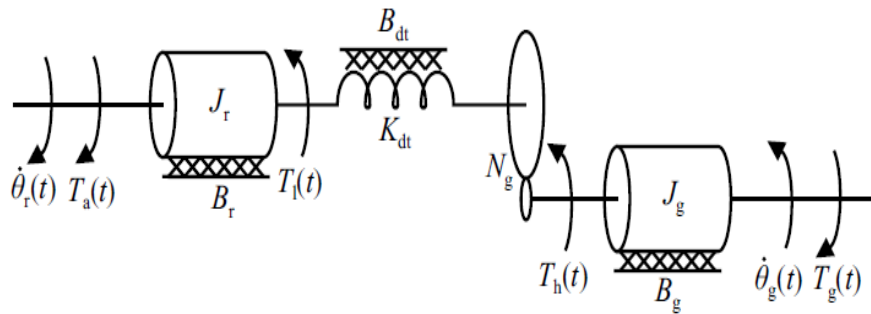
Figure 2.4: Drive-train model divided into four components

The inertia of the low-speed shaft also includes the inertia of the rotor, while the friction component includes bearing frictions. The dynamics of the low-speed shaft is:

$$J_r\ddot{\theta}_r(t) = T_a(t) - T_l(t) - B_r\dot{\theta}_r(t) \quad [\text{Nm}] \tag{2.41}$$

where:
B_r is the viscous friction of the low-speed shaft [Nm/(rad/s)]
J_r is the moment of inertia of the low-speed shaft [kg m^2]
$T_l(t)$ is the torque acting on the low-speed shaft [Nm].
$\theta_r(t)$ is the angle of the low-speed shaft [rad].
$T_a(t)$ is the aerodynamic torque applied to the rotor [Nm].

The inertia of the high-speed shaft also includes the inertia of the gearbox and the generator rotor. The friction coefficient covers bearing and gear frictions. The dynamics of the high-speed shaft is:

$$J_g\ddot{\theta}_g(t) = T_h(t) - T_g(t) - B_g\dot{\theta}_g(t) \quad \text{[Nm]} \tag{2.42}$$

where:
B_g is the viscous friction of the high-speed shaft [Nm/(rad/s)]
J_g is the moment of inertia of the high-speed shaft [kg m^2]
$T_g(t)$ is the generator torque [Nm]
$T_h(t)$ is the torque acting on the high-speed shaft [Nm]
$\theta_g(t)$ is the angle of the high-speed shaft [rad]

The remaining part of the gearbox modeling is to apply a gear ratio, as defined below:

$$T_h(t) = \frac{T_l(t)}{N_g} \quad \text{[Nm]} \tag{2.43}$$

where:
N_g is the drive train gear ratio [.]

The torsion of the drive train is modeled using a torsion spring and a friction coefficient model, described according to:

$$T_l(t) = K_{dt}\theta_\Delta(t) + B_{dt}\dot{\theta}_\Delta(t) \quad \text{[Nm]} \tag{2.44}$$

$$\theta_\Delta(t) = \theta_r(t) - \frac{\theta_g(t)}{N_g} \quad \text{[rad]} \tag{2.45}$$

where:
B_{dt} is the torsion damping coefficient of the drive train [Nm/(rad/s)]
K_{dt} is the torsion stiffness of the drive train [Nm/rad]
$\theta_\Delta(t)$ is the torsion angle of the drive train [rad]

With the exception of the torsion angle, $\theta_\Delta(t)$, absolute angles of the shafts are not of interest for modeling the drive train dynamics. Therefore, the replacement $\omega(t) = \dot{\theta}(t)$ is utilized in the following rewriting, where a state space model of the drive train is pursued. The states of the model are $\omega_r(t)$, $\omega_g(t)$, and $\theta_\Delta(t)$. First, Eq. (2.45) is substituted into Eq. (2.44) to obtain:

$$T_l(t) = K_{dt}\theta_\Delta(t) + B_{dt}\left(\omega_r(t) - \frac{\omega_g(t)}{N_g}\right) \quad \text{[Nm]} \tag{2.46}$$

where:
$\omega_g(t)$ is the generator speed [rad/s]
Substituting Eq. (2.46) into Eq. (2.41) results in Eq. (2.47). A similar approach is used to derive Eq. (2.48); however, in this case Eq. (2.46) first has to be substituted into Eq. (2.43) before inserting it in Eq. (2.42). Lastly, Eq. (2.45) is differentiated to obtain Eq. (2.49).
Three first order differential equations have been derived in this section in order to describe the behavior of the drive train

$$J_r\dot{\omega}_r(t) = T_a(t) - K_{dt}\theta_\Delta(t) - (B_{dt} + B_r)\,\omega_r(t) + \frac{B_{dt}}{N_g}\omega_g(t) \tag{2.47}$$

$$J_g \dot{\omega}_g(t) = \frac{K_{dt}}{N_g} \theta_\Delta(t) + \frac{B_{dt}}{N_g} \omega_r(t) - \left(\frac{B_{dt}}{N_g^2} + B_g \right) \omega_g(t) - T_g(t) \tag{2.48}$$

$$\dot{\theta}_\Delta(t) = \omega_r(t) - \frac{1}{N_g} \omega_g(t) \tag{2.49}$$

2.3.3 Benchmark Drive Train Model

The next Drive Train model proposed in the benchmark [37] follows the same line of the previous one.

It is basically the same model proposed in [11] with the addition of the efficiency of the drive train η_{dl} which gives a more realistic approach to the model.

The drive train state-space model is the following.

$$\dot{x} = \mathrm{A}x + \mathrm{B}u \tag{2.50}$$

$$y = \mathrm{C}x \tag{2.51}$$

where:

$$x = [\omega_r \ \omega_g \ \theta_\Delta]^T, \tag{2.52}$$

$$u = [T_a \ T_g]^T, \tag{2.53}$$

$$y = [\omega_r \ \omega_g]^T \tag{2.54}$$

$$
\begin{aligned}
\mathrm{A} &= \begin{bmatrix} -\frac{B_{dt}+B_r}{J_r} & \frac{B_{dt}}{N_g J_r} & \frac{-K_{dt}}{J_r} \\ \frac{\eta_{dl} B_{dt}}{N_g J_g} & \frac{-\frac{\eta_{dl} B_{dt}}{N_g^2} - B_g}{J_g} & \frac{\eta_{dl} K_{dt}}{N_g J_g} \\ 1 & \frac{-1}{N_g} & 0 \end{bmatrix}, \\
\mathrm{B} &= \begin{bmatrix} \frac{1}{J_r} & 0 \\ 0 & -\frac{1}{J_g} \\ 0 & 0 \end{bmatrix} \\
C &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
\end{aligned}
\tag{2.55}
$$

Table 2.2: Parameters of the state-space drive-train model

Parameter	Units	Description
J_r	kg m^2	Moment of inertia of the low speed shaft (rotor)
J_g	kg m^2	Moment of inertia of the high speed shaft (generator)
K_{dt}	N m/rad	Torsion stiffness of the drive train
B_{dt}	N m/(rad/s)	Torsion damping coefficient of the drive train
B_g	N m/(rad/s)	Viscous friction of the high speed shaft (generator),
B_r	N m/(rad/s)	Viscous friction of the low speed shaft (rotor),
N_g	-	Gear ratio
η_{dl}	-	Efficiency of the drive train,

Table 2.3: Variables of the state-space drive-train model

Variable	Units	Description
$w_r(t)$	rad/s	Rotor speed
$w_g(t)$	rad/s	Generator speed
θ_Δ	rad	Torsion angle of the drive train

2.3.4   Proposed Drive Train Model

After reviewing the previous three models, the proposed model is the two shafts model described in equations (2.47- 2.49). The proposed model includes the biggest amount of relations between variables and parameters of all the models analyzed. It has the most complete description of the wind turbine dynamics which will be very useful for further fault diagnosis analysis.

2.4   Pitch Subsystem

Following [32], the hydraulic pitch system is modeled as a second order transfer function between the pitch angle β and the pitch reference angle β_r

$$\frac{\beta(s)}{\beta_r(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.56}$$

There is a transfer function associated to each of the three pitch systems. In normal operation, the system has the following parameters $\zeta = 0.6$ and $\omega_n = 11.11$ leading to an under-damped system.

Also, constraints on the values of pitch angles and rate are implemented. The pitch angle values are restricted to the interval [-2, 90] deg and pitch rate is restricted to the interval [-8, 8] deg/s.

Similarly to this approach, the pitch system is modeled by [11] with a same transfer function to each of the three pitch systems in the wind turbine. However, in this model a time delay corresponding to the communication delay to the pitch actuator is added to the second order transfer function described as:

$$\frac{\beta(s)}{\beta_r(s)} = \frac{e^{(-t_d s)}\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.57}$$

A different pitch actuator model is suggested in [13], here the model is a first-order dynamic system with saturation in the amplitude and derivative of the output signal.

$$\dot{\beta} = -\frac{1}{\tau}\beta + \frac{1}{\tau}\beta_d \tag{2.58}$$

where: β is the pitch angle and β_d is the desired pitch angle.

Applying the Laplace transform we obtain the corresponding first order transfer function.

$$\frac{\beta(s)}{\beta_d(s)} = \frac{1/\tau}{\left(s + 1/\tau\right)} \tag{2.59}$$

In this model, the pitch angle varies in the interval [-2,30] deg and the rate in the interval [-10,10] deg/s.

2.4.1   Neglected Dynamics and Pitch Models Comparison

Three different model designs were presented. Two of them modeled the pitch actuator of the wind turbine by second order transfer functions, [11] and [32].

The difference between them is that in [11], the second order transfer function includes an exponential function to model the effect of a communication delay.

The first two models presented [11] and [32], considered an under-damped behavior of the pitch actuator.

In the third model presented [11], an over-damped dynamic was modeled using a first order differential equation.

Between the model [32] and [13], there are differences in the values of the intervals of pitch angles and rates.

2.4.2   Chosen Model

After the model comparison and the dynamics considered and neglected by each one of them, the model selected to represent the pitch actuator is the following:

$$\frac{\beta(s)}{\beta_r(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.60}$$

We consider that the time delay of the system is not very high in this case and can be neglected still obtaining a realistic representation of the system.

The first order system presented in [13] is very simple and we consider that an over-damped dynamic does not represent in a realistic way the behavior of a pitch actuator.

2.5   Tower Subsystem

The tower of a wind turbine, as any flexible structure, exhibits many vibration modes. Some oscillatory movements inherent to these modes are illustrated in Figure 2.5. Particularly, simple models are very helpful for a comparative analysis of different control strategies and for the controller design, whereas the un-modeled dynamics can be treated as uncertainties. For this reason, the model presented here will include the first mode of tower bending and the first mode of flapping.

In this thesis, it is assumed that the thrust on the rotor acts in the direction of the wind speed. In reality this is not completely the situation, there is also the side-to-side mode of tower bending which is produced by forces from the blades and a counter torque from the drive train and generator, which make the tower to swing sideways as well [11].

The model presented on Figure 2.6 has two degrees of freedom, which are the axial tower bending and the flapping. For this model the following generalized coordinates are adopted:

$$q = [y_t \ \ \zeta \ ]^T, \tag{2.61}$$

where y_t is the axial displacement of the nacelle and ζ is the angular displacement out of the plane of rotation of the blades.

To obtain the model of the tower, the Lagrange energy method is used. For mechanical structures having few degrees of freedom, the Lagrange equation

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{q}_i}\right) - \frac{\partial E_k}{\partial q_i} + \frac{\partial E_d}{\partial \dot{q}_i} + \frac{\partial E_p}{\partial q_i} = Q_i \tag{2.62}$$

offers a systematic procedure to derive mathematical models.

The energy terms E_k, E_d and E_p can be written as:

(a) Torsion

(b) Flapwise

(c) Tower bending

(d) Edgewise
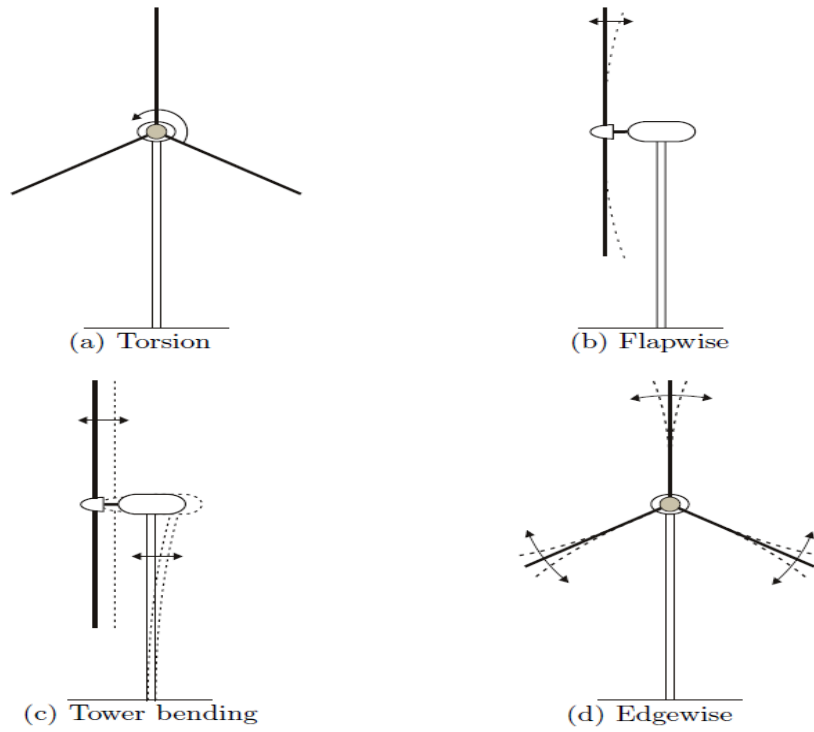
Figure 2.5: Mode shapes for horizontal-axis wind turbines.
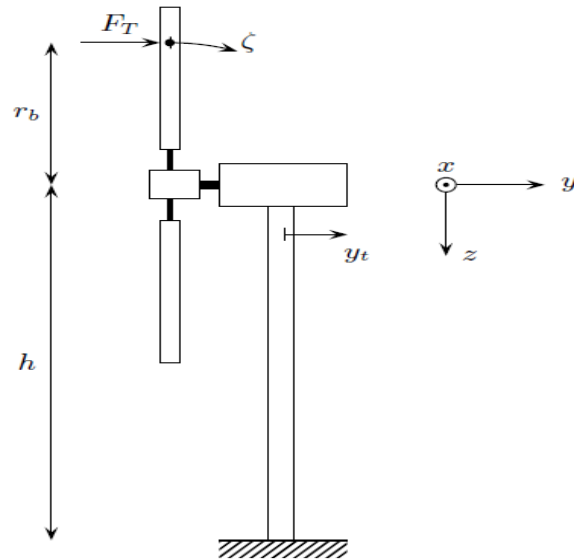
Figure 2.6: Schematic diagram of the tower.

$$E_k = \frac{m_t}{2}\dot{y}_t^2 + \frac{N}{2}m_b\left(\dot{y}_t + r_b\dot{\zeta}\right)^2 \tag{2.63}$$

$$E_d = \frac{B_t}{2}\dot{y}_t^2 + \frac{N}{2}B_b\left(r_b\dot{\zeta}\right)^2 \tag{2.64}$$

$$E_p = \frac{K_t}{2} y_t{}^2 + \frac{N}{2} K_b (r_b \zeta)^2 \tag{2.65}$$

The vector of generalized loads is:

$$Q = [N F_T \ \ N F_T r_b]^T. \tag{2.66}$$

For the first generalized coordinate $q_1 = y_t$, the Lagrange terms are:

$$\frac{\partial E_k}{\partial \dot{q}_1} = \frac{\partial E_k}{\partial \dot{y}_t} = m_t \dot{y}_t + N m_b \dot{y}_t + N m_b r_b \dot{\zeta}, \tag{2.67}$$

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{y}_t} \right) = (m_t + N m_b) \ddot{y}_t + N m_b r_b \ddot{\zeta}, \tag{2.68}$$

$$\frac{\partial E_k}{\partial y_t} = 0 \tag{2.69}$$

$$\frac{\partial E_d}{\partial \dot{q}_1} = \frac{\partial E_d}{\partial \dot{y}_t} = B_t \dot{y}_t \tag{2.70}$$

$$\frac{\partial E_p}{\partial q_1} = \frac{\partial E_p}{\partial y_t} = K_t y_t \tag{2.71}$$

$$Q_1 = N F_T \tag{2.72}$$

Substituting the equations (2.67)-(2.72) in the Lagrange equation (2.62), the dynamic of the tower bending is:

$$(m_t + N m_b) \ddot{y}_t + N m_b r_b \ddot{\zeta} + B_t \dot{y}_t + K_t y_t = N F_T \tag{2.73}$$

For the second generalized coordinate $q_2 = \zeta$, the Lagrange terms are the following:

$$\frac{\partial E_k}{\partial \dot{\zeta}} = \frac{N}{2} m_b \left(2 r_b \dot{y}_t + 2 r_b^2 \dot{\zeta} \right) \tag{2.74}$$

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{\zeta}} \right) = N m_b r_b \ddot{y}_t + N m_b r_b^2 \ddot{\zeta} \tag{2.75}$$

$$\frac{\partial E_k}{\partial \zeta} = 0 \tag{2.76}$$

$$\frac{\partial E_d}{\partial \dot{\zeta}} = N B_b r_b^2 \dot{\zeta} \tag{2.77}$$

$$\frac{\partial E_p}{\partial \zeta} = N K_b r_b^2 \zeta \tag{2.78}$$

$$N m_b r_b \ddot{y}_t + N m_b r_b^2 \ddot{\zeta} + N B_b r_b^2 \dot{\zeta} + N K_b r_b^2 \zeta = N F_T r_b \tag{2.79}$$

After simplifying equation (2.79), the dynamic of the blades flap-wise is:

$$m_b r_b \ddot{y}_t + m_b r_b^2 \ddot{\zeta} + B_b r_b^2 \dot{\zeta} + K_b r_b^2 \zeta = F_T r_b \tag{2.80}$$

Writing equations (2.73) and (2.80) in matrix form the following equation is obtained:

$$\begin{bmatrix} m_t + N m_b & N m_b r_b \\ m_b r_b & m_b r_b^2 \end{bmatrix} \begin{bmatrix} \ddot{y}_t \\ \ddot{\zeta} \end{bmatrix} + \begin{bmatrix} B_t & 0 \\ 0 & B_b r_b^2 \end{bmatrix} \begin{bmatrix} \dot{y}_t \\ \dot{\zeta} \end{bmatrix} + \begin{bmatrix} K_t & 0 \\ 0 & K_b r_b^2 \end{bmatrix} \begin{bmatrix} y_t \\ \zeta \end{bmatrix} = \begin{bmatrix} N F_T \\ F_T r_b \end{bmatrix} \tag{2.81}$$

A mechanical system of arbitrary complexity can be described by the equation of motion

$$M\ddot{q} + D\dot{q} + Kq = Q(\dot{q}, q, t, u) \tag{2.82}$$

where M, D and K are the mass, damping and stiffness matrices and Q is the vector of forces acting on the system. The system obtained in equation (2.81) has the same structure of equation (2.82).

From the equation (2.82), an state-space model can be obtained making the following operations

$$\ddot{q} = -\mathrm{M}^{-1}\mathrm{K}q - \mathrm{M}^{-1}\mathrm{D}\dot{q} + \mathrm{M}^{-1}\mathrm{Q} \tag{2.83}$$

$$\ddot{q} = \begin{bmatrix} -\mathrm{M}^{-1}\mathrm{K} & -\mathrm{M}^{-1}\mathrm{D} \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \mathrm{M}^{-1}\mathrm{Q} \tag{2.84}$$

The state vector is:

$$x = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}^T = \begin{bmatrix} y_t & \zeta & \dot{y}_t & \dot{\zeta} \end{bmatrix}^T \tag{2.85}$$

and therefore,

$$\dot{x} = \begin{bmatrix} \dot{y}_t & \dot{\zeta} & \ddot{y}_t & \ddot{\zeta} \end{bmatrix}^T \tag{2.86}$$

The system input is $u = F_T$, while vector Q is:

$$\mathrm{Q} = \begin{bmatrix} N \\ r_b \end{bmatrix}. \tag{2.87}$$

The output is:

$$y = \begin{bmatrix} \ddot{y}_t & \zeta \end{bmatrix}^T. \tag{2.88}$$

The state space model of the tower is:

$$\begin{cases} \dot{x} = \mathrm{A}x + \mathrm{B}u, \\ y = \mathrm{C}x, \end{cases} \tag{2.89}$$

and the matrices A, B and C are:

$$A = \begin{bmatrix} 0_2 & I_2 \\ -\mathrm{M}^{-1}\mathrm{K} & -\mathrm{M}^{-1}\mathrm{D} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{K}{m_t} & \frac{KN}{r_b m_t} & -\frac{B_t}{m_t} & \frac{Nr_b B_b}{m_t} \\ \frac{K}{r_b m_t} & \frac{-K(m_t+Nm_b)}{m_b r_b^2 m_t} & \frac{B_t}{r_b m_t} & \frac{-(m_t+Nm_b)B_b}{m_b m_t} \end{bmatrix},$$

$$B = \begin{bmatrix} 0_{2\times1} \\ \mathrm{M}^{-1}\mathrm{Q} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{-N}{r_b m_t} + \frac{m_t+Nm_b}{m_b r_b m_t} \end{bmatrix}, \tag{2.90}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

2.5.1   Tower Model Simplification

In order to obtain a simplified model of the tower for further fault diagnosis analysis, the blades are assumed to be stiff but in fact the blades are flexible as explained before. This simplification eliminates all the bending modes of the blades and transfers all forces acting on blades directly to the tower.

As explained later in this chapter, the tower top acceleration is typically the only measured variable in the tower model. Therefore, the movement of the tower is now described by a linear displacement of the tower top [11]. An illustration of this model is given in Figure 2.7.

Using a spring-damper terminology, the tower model is rewritten as:

$$M_t\ddot{y}_t(t) = F_{th}(t) - B_t\dot{y}_t(t) - K_t y_t(t) \tag{2.91}$$

Table 2.4: Parameters for the tower subsystem model

Parameter	Units	Description
m_t	kg	Mass of the tower and nacelle
m_b	kg	Mass of each blade
K_t	N m/rad	Stiffness of the tower
K_b	N m/rad	Stiffness of each blade
B_t	N m/(rad/s)	Damping coefficient of the tower
B_b	N m/(rad/s)	Damping coefficient of the blade
N	-	Number of blades
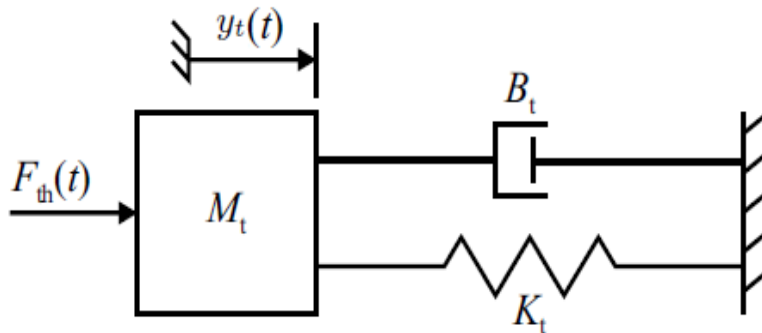
Figure 2.7: The movement of the flexible tower is modeled using a spring-damper system.

where:
B_t is the tower damping coefficient [N/(m/s)].
K_t is the tower torsion coefficient [N/m].
M_t is the top mass of the tower [kg].
$y_t(t)$ is the displacement of the nacelle from its equilibrium position [m].

2.5.2 Blade Root Moment Dynamics

This section is based on the work of [28] from which the Blade Root Moment Dynamics can be obtained.

It is assumed that the wind speed signal, when it acts on the whole rotor blade, causes blade root loads that are similar to those that arise when a rotating rotor blade samples a turbulent wind field, affected by wind shear and tower shadow. This concept allows for describing the wind influence on the blade (root) loads via a single input signal while yet taking into account the properties of the rotationally sampled wind field.

The aerodynamic conversion in the simplified model is based on linearized BEMtheory [42]; dynamic wake effects and unsteady aerodynamics are not taken into account. The BEM-based aerodynamic conversion characteristics are translated into multipliers that map a variation in the flapwise relative wind speed v_{flap_i} to variations in the flap and leadwise blade root moments and forces (aerodynamic gains). Aerodynamic gains are also derived for the linearized influence of a variation in the pitch angle. The pitch angle variation β_i and relative wind speed variation v_{flap_i} for the ith blade thus cause variations in the aerodynamic loads on the blade root described by:

$$M_{B,i}(t) = k_1 \left(v_r(t) - \frac{dy(t)}{dt} + \frac{9R_b}{8H} \frac{dy(t)}{dt} \sin(\psi_i(t)) \right) + k_2 \beta_i(t) \tag{2.92}$$

where:

$M_{B,i}(t)$ is the blade root moment on blade i [Nm].
$\psi_i(t)$ is the azimuth angle of blade i [rad].
$v_r(t)$ is the wind speed [m/s].
$\frac{dy(t)}{dt}$ is the translation speed of the nacelle from its equilibrium position [m/s].
R_b is the distance from the hub to where the thrust acts on the blade [m].
H is the hub height [m].

The gains k_1 and k_2 are derived from the power and thrust coefficient data under the assumption of equal aerodynamic efficiency over the rotor radius in a chosen working point.

For azimuth angle ψ_i equal to 0, the ith blade is in the horizontal position while it is rotating downward. The azimuth angle of the blade equals the rotor azimuth angle θ_r, which means

$$\theta_r = \psi_1(t) \quad [rad] \tag{2.93}$$

The rest of the azimuth blade angles can be calculated as shown below:

$$\psi_2(t) = \theta_r + \frac{2}{3}\pi \quad [rad] \tag{2.94}$$

$$\psi_3(t) = \theta_r + \frac{4}{3}\pi \quad [rad] \tag{2.95}$$

2.6 Aerodynamic Model

In this section, basic aerodynamic principles exploited by wind turbines are described, and a model describing the transfer from wind energy to rotational motion of the rotor is presented.

The power available from the wind passing through the entire rotor swept area can be expressed as [13]:

$$P_w(t) = \frac{1}{2}\rho A v_r^3(t) \quad [W] \tag{2.96}$$

where:
$P_w(t)$ is the power available from the wind.
A is the rotor swept area[m^2]
$v_r(t)$ is the rotor effective wind speed [m/s].
ρ is the air density, which is assumed to be constant [kg/m^3].

From the available power in the wind, the power on the rotor is given based on the power coefficient, $C_p(\lambda(t), \beta(t))$, which depends on the tip-speed ratio and the pitch angle. The C_p coefficient is not expressed as a mathematical function, but has to be looked up in a table. The C_p-surface for the wind turbine used in the FDI benchmark [32], is provided by kk-electronic a/s and is shown in the left subplot of Figure 2.8. Notice that the C_p-description implies that the aerodynamic model is static, which is a simplification.

The power captured by the rotor is:

$$P_a(t) = P_w(t)C_p(\lambda(t), \beta(t)) \quad [W] \tag{2.97}$$

where:
$P_a(t)$ is the power captured by the rotor [W].
$C_p(\lambda(t), \beta(t))$ is the power coefficient [-].
$\beta(t)$ is the pitch angle [deg].
$\lambda(t)$ is the tip-speed ratio [-].

The tip-speed ratio is defined as the ratio between the tip speed of the blades and the rotor effective wind speed:

$$\lambda(t) = \frac{\omega_r(t)R}{v_r(t)} \tag{2.98}$$

where:
$\omega_r(t)$ is the rotor speed [rad/s].
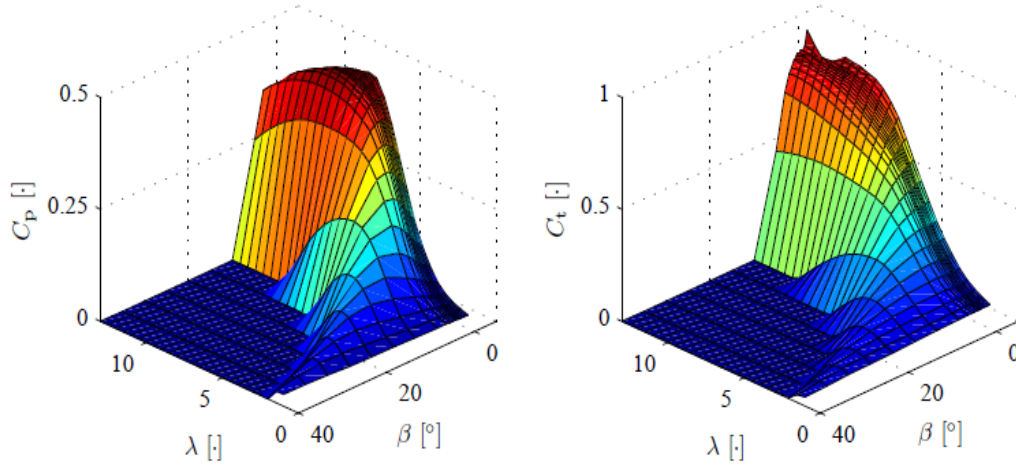
Figure 2.8: The C_p and C_t-coefficients as function of the pitch angle and the tip-speed ratio. Notice that negative values have been set to zero.

The aerodynamic torque applied to the rotor defined in [11], can be expressed as:

$$T_a(t) = \frac{P_a(t)}{\omega_r(t)} \qquad \text{[Nm]} \tag{2.99}$$

Substituting the equations (2.96) and (2.97), in (2.99) the torque applied to the rotor is expressed as:

$$T_a(t) = \frac{1}{2\omega_r(t)}\rho A v_r^3(t) C_p\left(\lambda(t), \beta(t)\right) \qquad \text{[Nm]} \tag{2.100}$$

where: $T_a(t)$ is the aerodynamic torque applied to the rotor [Nm].

The wind acting on the rotor of the wind turbine also results in a thrust on the rotor. This thrust is calculated as shown below [13].

$$F_t(t) = \frac{1}{2}\rho A v_r^2(t) C_t(\lambda(t), \beta(t)) \qquad \text{[N]} \tag{2.101}$$

where:
$C_t(\lambda(t), \beta(t))$ is the thrust coefficient [-].
$F_t(t)$ is the thrust exerted by the wind on the rotor [N].

To use the aerodynamic model when the wind speed is assumed to be non-identical on the three blades, the equations have to take into account different blade effective wind speeds. This is accomplished by averaging the thrust and torque introduced at each of the three blades, as illustrated below

$$F_{t,1}(t) = \frac{1}{2}\rho A v_r^2(t) C_t(\lambda(t), \beta_1(t)) \qquad \text{[N]} \tag{2.102}$$

$$F_{t,2}(t) = \frac{1}{2}\rho A v_r^2(t) C_t(\lambda(t), \beta_2(t)) \qquad \text{[N]} \tag{2.103}$$

$$F_{t,3}(t) = \frac{1}{2}\rho A v_r^2(t) C_t(\lambda(t), \beta_3(t)) \qquad \text{[N]} \tag{2.104}$$

$$T_{a,1}(t) = \frac{1}{2\omega_r(t)}\rho A v_r^3(t) C_p\left(\lambda(t), \beta_1(t)\right) \qquad \text{[Nm]} \tag{2.105}$$

$$T_{a,2}(t) = \frac{1}{2\omega_r(t)}\rho A v_r^3(t) C_p\left(\lambda(t), \beta_2(t)\right) \qquad \text{[Nm]} \tag{2.106}$$

$$T_{a,3}(t) = \frac{1}{2\omega_r(t)}\rho A v_r^3(t) C_p\left(\lambda(t), \beta_3(t)\right) \qquad [\text{Nm}] \tag{2.107}$$

$$T_a(t) = \frac{1}{3}\sum_{i=1}^{3} T_{a,i}(t) \qquad [\text{Nm}] \tag{2.108}$$

Substituting equations (2.105), (2.106) and (2.107) in (2.108), the following equation for the aerodynamic torque is obtained:

$$F_t(t) = \frac{1}{3}\sum_{i=1}^{3} F_{t,i}(t) \qquad [\text{N}] \tag{2.109}$$

where:
$F_{t,i}(t)$ is the thrust exerted by the wind on Blade i [N].
$T_{a,i}(t)$ is the aerodynamic torque applied to the rotor by Blade i [Nm].

It is assumed that the thrust $F_{t,i}(t)$ exerted on Blade i attacks where the thrust components towards the hub and towards the blade tip are equal. Since the thrust is dependent on R^2 the following equation can be set up to calculate the point where the thrust attacks:

$$\begin{aligned}\int\limits_{0}^{r_b} r^2 dr &= \int\limits_{r_b}^{R} r^2 dr \\ r_b &= 2^{-1/3}R \qquad [\text{m}]\end{aligned} \tag{2.110}$$

where:
r_b is the distance from the hub to where the thrust acts on the blade [m].
R is the radius of the rotor [m].
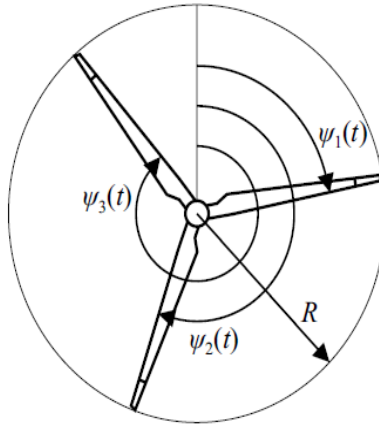
Figure 2.9: Sketch of a rotor and a wind turbine, showing parameters utilized in the wind model.

$\psi_i(t)$ is the azimuth angle of Blade i [rad].

Having determined where the thrust acts on the blades, it is possible to derive the torque which makes the tower move. This is a function of the azimuth angle of each blade and can be realized as a force acting on the tower at hub height, as shown below.

$$\begin{aligned}F_{th}(t) = F_{t,1}(t)\left(1 + \tfrac{r_t}{h}\cos\left(\psi_1(t)\right)\right) + F_{t,2}(t)\left(1 + \tfrac{r_t}{h}\cos\left(\psi_2(t)\right)\right) \\ + F_{t,3}(t)\left(1 + \tfrac{r_t}{h}\cos\left(\psi_3(t)\right)\right) \qquad [\text{N}]\end{aligned} \tag{2.111}$$

where:

$F_{t,i}(t)$ is the thrust acting on Blade i [N].
$F_{th}(t)$ is the force acting on the tower at hub height [N].
$F_{th,i}(t)$ is the force transferred to the tower from Blade i at hub height [N].

The basic aerodynamic principles of a wind turbine have been described, and models for the aerodynamic torque and the aerodynamic thrust acting on the rotor have been set up.

2.7 Power Subsystem Model

The power system as modeled in [32] and [11] is considered a standard first order system which is very similar in both cases.

Electric power is generated by the generator, and to enable variable-speed operation, currents in the generator are controlled using power electronics. Therefore, power electronic converters interface the wind turbine generator output with the utility grid. It is assumed that the converter consists of four similar units sketched in Figure 2.10, each having an internal controller. These units together load the generator with a certain torque, which depends on the currents drawn from the generator. Since torque and electric power are the only variables of interest in the simplified model of the energy conversion system, currents and voltages are not considered at all.

Since the converter consists of several converters having equal characteristics, this section describes only one of these.
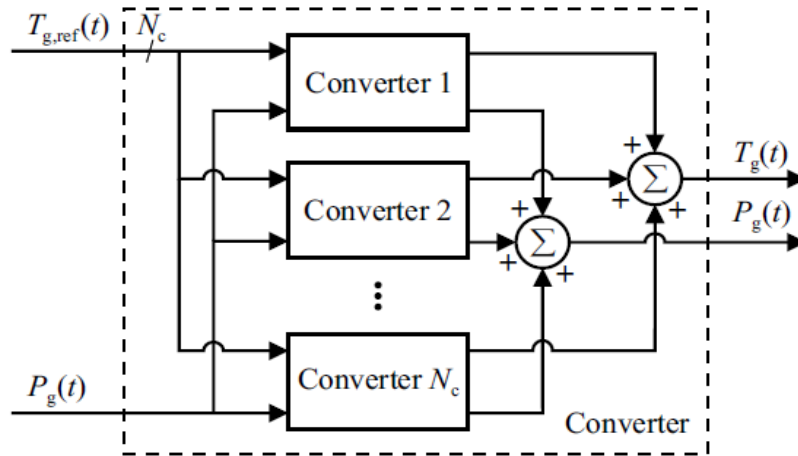
Figure 2.10: The converter consists of N_c units capable of loading the generator by a certain torque, specified by a torque reference.

On a system level of the wind turbine, the generator and converter dynamics can be modeled by a first order transfer function:

$$\frac{T_g(s)}{T_{g,ref}(s)} = \frac{1}{\tau_g s + 1} \tag{2.112}$$

where:
$T_{g,ref}(t)$ is the reference for the generator torque [Nm].
τ_g is the time constant of the first order system [s].

The power produced by the generator depends on the rotational speed of the rotor and of the applied load, as described in the equation below. The following equation explains the mechanical power since the electronic system is not modeled:

$$P_g(t) = \eta_g \omega_g(t) T_g(t) \quad [W] \tag{2.113}$$

where:
$P_g(t)$ is the power produced by the generator [W].
η_g is the efficiency of the generator [-].

In the wind turbine FDI benchmark [32], the parameter values for the efficiency and the time constant are $\eta_g = 0.98$ and $\tau_g = \frac{1}{50}$ respectively.

In this section, the generator and converter models have been presented. The generator converts mechanical energy into electric energy, while it is loaded by a torque originating from a converter, which has been described as a number of delayed first order systems.

Chapter 3

Structural Analysis

In this chapter the purpose is to perform a structural analysis of the wind turbine system in order to determine the analytical redundancy relations (ARRs), which can be used to detect and isolate faults in the system. Furthermore, the possible detectable and isolable faults and sensor fusion possibilities are determined by exploiting the ARRs.

Structural analysis is the analysis of the structural properties of models, i.e., properties which are independent of the actual values of the parameters [12]. It only represents the links between the variables and the parameters which result from the model and are thus independent of the form under which this model is expressed (quantitative or qualitative). The links are presented by a graph upon which the analysis of the structure will be performed.

Structural analysis will provide tools for several steps of the design of the fault diagnosis system: analysis of local redundancies, determination of extra sensors in order to improve diagnosis results, determination of computational sequences whose result is a residual (a conflict), evaluation of fault detectability and isolability, possibilities to implement fault tolerance actions.

From the point of view of structural analysis, the model of the system is considered as a set of constraints which apply to a set of variables some that are known and other that are unknown: -the sensors which are present in the process together with the control variables, give the subset of those variable whose values are known, the set of constraints is given by models of the components which constitute the system.

The term constraints refer to the fact that components impose some relations between the values of variables according their corresponding physical laws.

Let $R = \{r_1, r_2, \ldots, r_m\}$ be the set of the constraints which represent the system model and let $Z = \{z_1, z_2, \ldots, z_n\}$ be the set of the variables which contains three subsets:

Let $K = Y \cup U$ be the set of known variables: U is the subset of input variables, Y is the subset of the output variables and X is the subset of the unknown (non-measured) variables.

The structure of the model is described by the binary relation:

$$
\begin{aligned}
&S : R \times Z \to \{0,1\} \\
&where \\
&(r_i, z_j) \to \left\{ \begin{array}{ll} S\left(r_i, z_j\right) = 1 & iff \quad r_j \ applies \ to \ z_j \\ S\left(r_i, z_j\right) = 0 & otherwise \end{array} \right.
\end{aligned}
\tag{3.1}
$$

To summarize, The structural analysis is set up according to the procedure shown in the bullet list below [19].

Structural Relations: Determine the structural relations of the wind turbine model and organize the structural relations in an incidence matrix. An incidence matrix shows structural relations in the model and separates known from unknown variables.

Matching: Perform a matching of the unknown variables. A matching shows how the unknown variables can be estimated from the known variables using the constraints in the model. The matching procedure can be done using a graph or a matrix.

Analytical Redundancy Relations: Determine the analytical redundancy relations and sensor fusion possibilities.

3.1   System Sensors

To perform a structural analysis and determine the analytical redundant relations of the wind turbine system it is necessary to know all the measured variables of the system. In table 3.1 is listed the measured variables of the system model and characterized the measurements by the amount of noise on each of them.

Table 3.1: Available sensors

Sensor Type	Symbol	Unit	Noise Power
Anemometer - Wind speed at hub height	$v_{w,m}$	m/s	0.0071
Rotor Speed	$\omega_{r,m}$	rad/s	10^{-4}
Generator Speed	$\omega_{g,m}$	rad/s	$2 \cdot 10^{-4}$
Generator Torque	$\tau_{g,m}$	Nm	0.9
Generated Electrical Power	$P_{g,m}$	W	10
Pitch angle of ith Blade	$\beta_{i,m}$	deg	$1.5 \cdot 10^{-3}$
Azimuth angle low speed side	$\theta_{r,m}$	rad	10^{-3}
Blade root moment ith blade	$M_{B,i,m}$	Nm	10^3
Tower top acceleration (x and y directions) measurement	$\begin{bmatrix} \ddot{x}_{x,m} \\ \ddot{x}_{y,m} \end{bmatrix}$	m/s^2	$5 \cdot 10^{-4}$
Yaw error	$\Xi_{e,m}$	deg	$5 \cdot 10^{-2}$

From the structural relations of the wind turbine model an incidence matrix is builded. This incidence matrix shows structural relations in the model and separates known from unknown variables.

The next step is a matching procedure which shows how the unknown variables can be estimated from the known variables using the constraints in the model.

3.2   Structural Relations

In this section the mutual dependence of the variables must be described; hence, the basis for the structural relations is the functional equations set up in Chapter 3.

The structural relations only show the variables which are linked together while the system equations described in the modeling chapter explain how variables are linked together. To form the structural relations it is necessary to use the system equations of chapter 3. In addition to the system equations, the measurement equations, which can be deducted from section 4.1 which gives the information of all the measured variables of the system, also form some structural relations.

Besides the system and measurement equations there are some differential relations between the variables and their derivatives. These relations, called differential constraints, are special, since the following holds: a signal $x(t)$ cannot be computed from its derivative $\dot{x}(t)$, but $\dot{x}(t)$ can be computed from $x(t)$. This is called integral causality, and is due to the lack of knowledge about the initial condition, $x(0)$, as indicated in [30].

$$x(t) = \int_0^t \dot{x}(\tau)d\tau + x(0) \tag{3.2}$$

In this thesis the structural relations are obtained from the physics modeling of the wind turbine. Table 3.2, summarizes the set of constraints related with the physical equations 2 and the measurement equations which can be deducted from Section 3.1, together with equations formed with the control inputs, also form some structural relations.

An incidence matrix is a way to represent the structural relations by separating the known variables K from the unknown variables X. When using the structural relations in an incidence matrix it is necessary to distinguish between known variables and unknown variables, since the incidence matrix is used in the matching. The known variables are control signals, measured variables, and known parameters, while structural relations show the variables which are linked together. The unknown variables are unmeasured variables and unknown parameters of the system.

Table 3.2: Structural Relations Summary

Structural Relation	Unknown Variables	Known Variables
e_1	2.41	$f\left(\dot{\omega}_r(t), \omega_r(t), T_l(t), T_a(t)\right) = 0$
e_2	2.42	$f\left(\dot{\omega}_g(t), \omega_g(t), T_h(t), T_g(t)\right) = 0$
e_3	2.43	$f\left(T_h(t), T_l(t)\right) = 0$
e_4	2.44	$f\left(T_l(t), \theta_\Delta(t), \dot{\theta}_\Delta(t)\right) = 0$
e_5	2.45	$f\left(\theta_\Delta(t), \theta_r(t), \theta_g(t)\right) = 0$
e_6	2.49	$f\left(\dot{\theta}_\Delta(t), \omega_r(t), \omega_g(t)\right) = 0$
e_7	2.60**	$f\left(\ddot{\beta}_1(t), \dot{\beta}_1(t), \beta_1(t), \beta_{ref}(t)\right) = 0$
e_8	2.60**	$f\left(\ddot{\beta}_2(t), \dot{\beta}_2(t), \beta_2(t), \beta_{ref}(t)\right) = 0$
e_9	2.60**	$f\left(\ddot{\beta}_3(t), \dot{\beta}_3(t), \beta_3(t), \beta_{ref}(t)\right) = 0$
e_{10}	2.91	$f\left(\ddot{y}_t(t), \dot{y}_t(t), y_t(t), F_{th}(t)\right) = 0$
e_{11}	2.92**	$f\left(M_{B,1}(t), v_r(t), \dot{y}_t, \psi_1(t), \beta_1(t)\right) = 0$
e_{12}	2.92**	$f\left(M_{B,2}(t), v_r(t), \dot{y}_t, \psi_2(t), \beta_2(t)\right) = 0$
e_{13}	2.92**	$f\left(M_{B,3}(t), v_r(t), \dot{y}_t, \psi_3(t), \beta_3(t)\right) = 0$
e_{14}	2.93	$f\left(\theta_r(t), \psi_1(t)\right) = 0$
e_{15}	2.94	$f\left(\theta_r(t), \psi_2(t)\right) = 0$
e_{16}	2.95	$f\left(\theta_r(t), \psi_3(t)\right) = 0$
e_{17}	2.96	$f\left(P_w(t), v_r(t)\right) = 0$
e_{18}	2.97	$f\left(P_a(t), P_w(t)\right) = 0$
e_{19}	2.98	$f\left(\lambda(t), v_r(t), \omega_r(t)\right) = 0$
e_{20}	2.99	$f\left(T_a(t), P_a(t), \omega_r(t)\right) = 0$
e_{21}	2.101	$f\left(F_t(t), v_r(t)\right) = 0$
e_{22}	2.111	$f\left(F_{th}(t), F_t(t), \psi_1(t), \psi_2(t), \psi_3(t)\right) = 0$
e_{23}	2.112	$f\left(\dot{T}_g(t), T_g(t), T_{g,ref}(t)\right) = 0$
e_{24}	2.113	$f\left(P_g(t), T_g(t), \omega_g(t)\right) = 0$
Measurement Equations		
m_1	$v_{r,m}(t) = v_r(t)$	$f\left(v_{r,m}(t), v_r(t)\right) = 0$
m_2	$\omega_{r,m}(t) = \omega_r(t)$	$f\left(\omega_{r,m}(t), \omega_r(t)\right) = 0$
m_3	$\omega_{g,m}(t) = \omega_g(t)$	$f\left(\omega_{g,m}(t), \omega_g(t)\right) = 0$
m_4	$T_{g,m}(t) = T_g(t)$	$f\left(T_{g,m}(t), T_g(t)\right) = 0$
m_5	$P_{g,m}(t) = P_g(t)$	$f\left(P_{g,m}(t), P_g(t)\right) = 0$
m_6	$\beta_{1,m}(t) = \beta_1(t)$	$f\left(\beta_{1,m}(t), \beta_1(t)\right) = 0$
m_7	$\beta_{2,m}(t) = \beta_2(t)$	$f\left(\beta_{2,m}(t), \beta_2(t)\right) = 0$
m_8	$\beta_{3,m}(t) = \beta_3(t)$	$f\left(\beta_{3,m}(t), \beta_3(t)\right) = 0$
m_9	$M_{B,1,m}(t) = M_{B,1}(t)$	$f\left(M_{B,1,m}(t), M_{B,1}(t)\right) = 0$
m_{10}	$M_{B,2,m}(t) = M_{B,2}(t)$	$f\left(M_{B,2,m}(t), M_{B,2}(t)\right) = 0$
m_{11}	$M_{B,3,m}(t) = M_{B,3}(t)$	$f\left(M_{B,3,m}(t), M_{B,3}(t)\right) = 0$
m_{12}	$\ddot{x}_m(t) = \ddot{x}(t)$	$f\left(\ddot{x}_m(t), \ddot{x}(t)\right) = 0$
m_{13}	$\ddot{y}_m(t) = \ddot{y}(t)$	$f\left(\ddot{y}_m(t), \ddot{y}(t)\right) = 0$
m_{14}	$\theta_{r,m}(t) = \theta_r(t)$	$f\left(\theta_{r,m}(t), \theta_r(t)\right) = 0$
Control Inputs		
c_1	$\beta_{ref,m}(t) = \beta_{ref}(t)$	$f\left(\beta_{ref,m}(t), \beta_{ref}(t)\right) = 0$
c_2	$T_{g,ref,m}(t) = T_{g,ref}(t)$	$f\left(T_{g,ref,m}(t), T_{g,ref}(t)\right) = 0$

The incidence matrix of the wind turbine system considered in this thesis is showed on Figures 3.1, 3.2 and 3.3.

Structural Eq.	θr=ψ1	ψ2	ψ3	θg	dωr/dt	dωg/dt	θΔ	dθΔ/dt	ωr	ωg(t)	λ(t)	Th(t)	Tl(t)	Ta(t)	Tg(t)	dTg(t)/dt	Tg_ref(t)	βref(t)	β1	β2	β3
																		Unknown Variables			
e1									×					×	×						
e2										×											
e3																					
e4																					
e5	×											×	×			×					
e6				×			×		×	×		×	×	×							
e7							×	×													
e8								×													
e9						×															
e10					×																
e11	×	×																			
e12		×																			
e13			×																		×
e14	×		×															×	×		
e15	×	×																×		×	
e16	×																	×			
e17												×									
e18									×		×										
e19									×	×				×							
e20									×		×										
e21															×						
e22																×					
e23															×	×	×				
e24										×				×	×						
Measurement Eq.																					
m1											×										
m2									×												
m3										×											
m4														×							
m5															×						
m6																			×		
m7																				×	
m8																					×
m9																					
m10																					
m11																					
m12																					
m13																					
m14	×																				
Control Inputs																					
c1																	×				
c2																		×			

Figure 3.1: Incidence Matrix

Figure 3.2: Incidence Matrix

Column labels (right side, **Known Variables**):

vr(t), Pa(t), ψ1, ψ2, ψ3, Pg(t), v_m, ωr_m, ωg_m, Tg_m, Pg_m, β1_m, β2_m, β3_m, MB1_m, MB2_m, MB3_m, d²xm/dt², d²ym/dt², θr_m, βref, Tg_ref

Figure 3.3: Incidence Matrix

3.3 Matching

A matching of the unknown variables is required to identify the monitorable part of the system, i.e. the subsystem in which faults can be detected and isolated. This part of the system must be observable to be in the over-constraint part of the system (C, Z). This means that there should be more constraints describing this part of the system than unknown variables [30].

The matching of the unknown variables is performed as shown in Figure , where the unknown variables are matched from the known variables using the structural relations. These relations are Analytical Redundant Relations (ARRs) and can be used for fault detection and fault identification purposes and will be detailed later in this chapter.

The process of obtaining ARRs from the matching procedure it is explained in Figure 3.4 where it is illustrated with the generator and converter subsystem, characterized in equation (2.112). In this example it is showed how the links are found between known variables (measured variables, controlled inputs), its derivatives and the unknown variables. The matching procedure can be done using an incidence matrix. This process is repeated for the rest of the subsystems of the wind turbine to generate other analytical redundant relations.

Structural Relation	Unknown Variables			Known Variables	
	$T_g(t)$	$\frac{dI_g(t)}{dt}$	$T_{gref}(t)$	$T_{gm}(t)$	$T_{grefm}(t)$
e₁	X	X	X		
Measurement Equations					
m₄	X			X	
Control Inputs					
c₂			X		X

Link between known and unknown variable
Link between known variable and its derivative
Link between known variable and unknown variable in the structural relation

Figure 3.4: Matching example with the Generator/Converter subsystem model

From the structural relation (2.112) and using the incidence matrix to do the matching, the unknown variables are replaced by known variables obtaining in this case one analytical redundant relation corresponding to the generator and converter subsystem of the wind turbine.

$$\tau_g \frac{dT_{g,m}(t)}{dt} + T_{g,m}(t) = T_{g,\mathrm{ref}}(t) \tag{3.3}$$

With the same procedure illustrated before and using the incidence matrix of the wind turbine system showed in Figure 3.1, 3.2 and 3.3, are obtained another analytical redundant relations detailed in the next section.

3.4   Analytical Redundancy Relations

Analytical redundancy relations (ARRs) are defined as relations between known variables. ARRs can be derived combining the measurement model (known variables) with the process model (unknown variables). From the analytical redundancy relations it is possible to create residuals for detecting faults. These relations will be used in the fault diagnosis procedure to check the consistency between the observed and the predicted process behavior [12].

From the structural relation (2.113) it is obtained directly another ARR from the matching procedure explained before.

$$P_{g,m}(t) = \eta_g \omega_{g,m}(t) T_{g,m}(t) \tag{3.4}$$

From the structural relation (2.92) and doing the matching in the incidence matrix for the blade root moment dynamics for each blade, the next three ARRs are obtained:

$$M_{B,1,m}(t) = k_1 \left(v_{r,m}(t) - \frac{dy(t)}{dt} + \frac{9R_b}{8H} \frac{dy(t)}{dt} \sin(\theta_{r,m}(t)) \right) + k_2 \beta_{1,m}(t) \tag{3.5}$$

$$M_{B,2,m}(t) = k_1 \left(v_{r,m}(t) - \frac{dy(t)}{dt} + \frac{9R_b}{8H} \frac{dy(t)}{dt} \sin\left(\theta_{r,m}(t) + \frac{2\pi}{3} \right) \right) + k_2 \beta_{2,m}(t) \tag{3.6}$$

$$M_{B,3,m}(t) = k_1 \left(v_{r,m}(t) - \frac{dy(t)}{dt} + \frac{9R_b}{8H} \frac{dy(t)}{dt} \sin\left(\theta_{r,m}(t) + \frac{4\pi}{3} \right) \right) + k_2 \beta_{3,m}(t) \tag{3.7}$$

Doing the matching procedure it is also possible to find ARRs for the pitch subsystem, each ARR corresponds to each one of the blades.

$$\frac{d^2 \beta_{1m}(t)}{dt^2} + 2\zeta\omega_n \frac{d\beta_{1m}(t)}{dt} + \omega_n^2 \beta_{1m}(t) = \omega_n^2 \beta_{ref}(t) \tag{3.8}$$

$$\frac{d^2 \beta_{2m}(t)}{dt^2} + 2\zeta\omega_n \frac{d\beta_{2m}(t)}{dt} + \omega_n^2 \beta_{2m}(t) = \omega_n^2 \beta_{ref}(t) \tag{3.9}$$

$$\frac{d^2 \beta_{3m}(t)}{dt^2} + 2\zeta\omega_n \frac{d\beta_{3m}(t)}{dt} + \omega_n^2 \beta_{3m}(t) = \omega_n^2 \beta_{ref}(t) \tag{3.10}$$

Studying the steady state behavior of the equations (2.47), (2.48) and (2.49) which describe the drive train dynamics, another equations can be obtained which contain only known variables and therefore form extra ARRS.

In steady state the rotor and generator accelerations are zero. Therefore equation (2.48) is the following:

$$0 = \frac{K_{dt}}{N_g} \theta_\Delta(t) + \frac{B_{dt}}{N_g} \omega_r(t) - \frac{B_{dt}}{N_g^2} \omega_g(t) - B_g \omega_g(t) - T_g(t) \tag{3.11}$$

Multiplying (4.21) by $-Ng$ we obtain equation (4.22).

$$0 = -K_{dt}\theta_\Delta(t) - B_{dt}\omega_r(t) + \frac{B_{dt}}{N_g} \omega_g(t) + N_g B_g \omega_g(t) + N_g T_g(t) \tag{3.12}$$

In steady state it is observed that relation (4.23), between the rotor speed and generator speed is satisfied.

$$\omega_r(t) - \frac{1}{N_g} \omega_g(t) = 0 \tag{3.13}$$

Therefore equation (4.24) is also satisfied.

$$\theta_r(t) - \frac{1}{N_g} \theta_g(t) = K = \theta_\delta(t) \tag{3.14}$$

Substituting equations (4.23) and (4.24) in eq. (4.21) the following equation is obtained:

$$T_g(t) = K - B_g \omega_g(t) \tag{3.15}$$

Because the rotor and generator speed are both measured variables, equation (4.23) can form an ARR and is written as

$$\omega_{r,m}(t) = \frac{1}{N_g} \omega_{g,m}(t) \tag{3.16}$$

And from equation (4.25) can be obtained another ARR because the generator's torque and the rotor speed are measured variables. This ARR is written as

$$T_{g,m}(t) = K - B_{g,m} \omega_g(t) \tag{3.17}$$

Once all the ARRs are deduced from the structural relations, a table listing all the ARRs with it's respective measured variables is builded. This is illustrated in Table (3.3).

The aerodynamic torque can be estimated from the rotor speed and wind speed, also the rotor power coefficient C_p is calculated and obtained from FAST.

$$T_{a*}(t) = f\left(v_{r,m}(t), \omega_{r,m}(t)\right) \tag{3.18}$$

Table 3.3: ARRs Table

	$\omega_{g,m}$	$\omega_{r,m}$	T_{gm}	$P_{g,m}$	$\beta_{1,m}$	$\beta_{2,m}$	$\beta_{3,m}$	$M_{B,1,m}$	$M_{B,2,m}$	$M_{B,3,m}$	$\beta_r ef$	T_{g-ref}	T_{a*}
r_1	x		x	x									
r_2					x			x					
r_3						x			x				
r_4							x			x			
r_5		x										x	
r_6					x						x		
r_7						x					x		
r_8							x				x		
r_9	x		x										
r_{10}	x	x	x										x
r_{11}	x	x	x										x
r_{12}	x	x											

3.5 Fault Scenarios

In the benchmark [32] both sensor and actuator faults are considered. Most faults selected were motivated by research, both in the public domain and from proprietary sources. In this section, the faults are described and, when possible, provide sources to data motivating their selection.

3.5.1 Sensor Faults

Sensor faults include measurements that are stuck, scaled from the true values, or offset from the true values, as indicated in Table 4.1. When Fault 1 occurs, the blade root bending moment sensor at Blade 2 is scaled by a factor of 0.95. Fault 1 is present between 295s and 320s. Fault 2 results in an offset of -0.5m/s^2 on the tower top accelerometer in both the fore-aft and side-to-side directions. Fault 2 is present in the time period 75s to 100s. Accelerometers are notoriously difficult to keep calibrated; see, for example, [23] and [10]. Fault 3 causes the generator speed sensor to be scaled by a factor of 0.95. Fault 3 is present between 130s and 155s. Fault 4 results Blade 1 having a stuck pitch angle sensor, which holds a constant value of

1 deg. Fault 4 is active from 185s to 210s. While Fault 5 is occurring the generator power sensor is scaled with a factor of 1.1. Fault 5 is present in the time interval from 240s to 265s. Precedent for scaled generator power sensor errors can be found in [23]. Fault 6 models a bit error in the low speed shaft encoder, which is another sensor fault documented in [23]. This bit error is modeled by randomly adding an offset to the measurement that corresponds to the bit on which the error is present. Fault 6 occurs from 295s to 320s.

3.5.2   Actuator faults

The primary actuators used by the turbine are the blade pitch drives, the generator torque, and the yaw drive. The two faults in the pitch actuators are Faults 7 and 8. These two faults are modeled by changing the parameters in the relevant pitch actuator model, and the same parameters are used as in [37]. Motivation for faults in pitch actuators is largely proprietary, but an example of a pitch actuator fault containing unexpected dynamics is given in [23]. In order to model the hydraulic power drop and increase of air content, the parameters in the transfer function are changed during these faults. Notice here that the hydraulic pressure drop is assumed to be abrupt, while the air content increases slowly. The two transfer function parameters for the pressure drop case are denoted $\omega_{n2} = 5.73$; $\zeta 2 = 0.45$ and the two parameters for the increased air content model are denoted $\omega_{n3} = 3.42$; $\zeta 3 = 0.9$. Fault 7 is introduced linearly from 350s to 370s, and full active from 370s to 390s and linearly outfaced from 390s to 410s. Fault 8 is active from 440s to 465s, and linearly introduced and outfaced within 1s both.

Fault 9 is an offset on the generated generator torque, which can be caused by an error in the initialization of the converter controller. This fault is modeled as in [37]; however, it was found based on the evaluation of the contribution to the FDI benchmark problem that the original offset value 100 Nm was to small to be detected (see [33]). Consequently the offset is increased to 1000 Nm. Fault 9 occurs from 495s-520s.

The last fault, which is the stuck yaw actuator, Fault 10, is modeled by setting the yaw angular velocity to zero rad/s independent of the value of the yaw error. Fault 10 is active in the time interval from 550s to 575s.

The faults are listed in the Table (4.1).

Table 3.4: Fault Scenarios

No.	Faults	Type
f_1	Blade root bending moment sensor	Scaling
f_2	Acceloremeter	Offset
f_3	Generator speed sensor	Scaling
f_4	Pitch angle sensor	Stuck
f_5	Generator power sensor	Scaling
f_6	Low speed shaft position encoder	Bit error
f_7	Pitch actuator	Abrupt change in dynamics
f_8	Pitch actuator	Slow change in dynamics
f_9	Torque offset	Offset
f_{10}	Yaw drive	Stuck drive

Analyzing the effects of the faults showed on Table (4.1) in the residuals listed on Table (3.3), the following fault signature matrix presented on Table (3.5) is obtained.

It has not been possible to find a suitable model of the yaw dynamics in the literature review. The structural analysis showed that the tower displacement and velocity were not observable therefore it was not possible to find an ARR to detect faults in this subsystem. After several simulation tests, the measurement equation which relates the azimuth angle with the rotor speed did not have a reasonable or expected behavior. For the reasons explained before, the benchmark fault scenarios 2, 6 and 10 were not considered in the fault diagnosis system implemented in this thesis.

Table 3.5: Theoretical signature matrix

	f_1			f_3	f_4			f_5	$f_{(7-8)}$			f_9
	$f_{M_{\mathrm{B},1,\mathrm{m}}}$	$f_{M_{\mathrm{B},2,\mathrm{m}}}$	$f_{M_{\mathrm{B},3,\mathrm{m}}}$		$f_{4-\beta1}$	$f_{4-\beta2}$	$f_{4-\beta3}$		$f_{(7-8)-PA1}$	$f_{(7-8)-PA2}$	$f_{(7-8)-PA3}$	
r_1				x				x				x
r_2	x				x				x			
r_3		x				x				x		
r_4			x				x				x	
r_5												x
r_6					x				x			
r_7						x				x		
r_8							x				x	
r_9				x								x
r_{10}												x
r_{11}				x								x
r_{12}				x								

3.6   Fault Isolation Techniques

The aim of fault isolation is to distinguish the fault that is present in the system between a set of possible faults. While a residual is enough to detect a fault, a set of residuals is required to isolate a fault [12]. A fault is isolable if it can be isolated from the rest of faults using a set of residuals. One of the main challenges in fault isolation is if the method used will be able to isolate some type of faults from other types. The answer to this challenge is related to: mainly, the structure of the model used to generate residuals and, in a certain way, to the detection test used.

Two distinct and parallel communities work on model-based diagnosis approaches: **FDI** in the control field and **DX** in the artificial intelligence field.

The FDI community focus in analytical redundancy approach while the DX focus in consistency-based logic approach.

There are some common principles between these two communities:
- The diagnostic process relies on an explicit model of the (normal) system behavior.
- The process detects faults from inconsistencies between the observations and the behavior predicted by the model.
- Fault isolation stands on interlinking the sets of components which underly every detected inconsistency. Even though the core principles are the same between these two communities, several concepts, assumptions and techniques differentiate them.

The DX approach relies upon a well founded and logically based theory for diagnosis of static systems. From a logical point of view, fault detection is performed through a consistency check, organized around the conflict concept. In this approach, fault localization or isolation is automatically derived from the conflict detection stage, which usually relies upon some kind of dependency-recording.

On the other hand, the FDI approach consider fault diagnosis as two separate tasks: fault detection and fault isolation based on generating and evaluating a set of analytical redundancy relations obtained off-line from elementary component models of the physical systems [43].

3.6.1   FDI fault isolation

Given a set of symptoms $s(k) = [s_1(k), ..., s_n(k)]$, and a set of considered faults $f_1, f_2, ... f_m$, the theoretical signature matrix can be defined binary codifying the effect or not of a fault in every symptom. This matrix has as many rows as symptoms and as many columns as faults are considered. The element $\sum_{ij}$ of this matrix is equal to 1 meaning that the jth fault appears in the expression of the ith symptom generator, otherwise is equal to 0. Then, fault isolation will consist in looking for the theoretical fault signature in the fault signature matrix that matches with the observed signature.

In the case of table (3.5), the following logical tests will allow to isolate the faults indicated below in table (3.6) without considering that noise or perturbations may cause detection errors, where ΔNr_i indicates an abnormal behavior of residual i^{th}:

Table 3.6: Fault Isolation FDI approach

Logical Test	Diagnostic
ΔNr_2	$f_{1-M,B,1,m}$
ΔNr_3	$f_{1-M,B,2,m}$
ΔNr_4	$f_{1-M,B,3,m}$
$\Delta Nr_1 \wedge \Delta Nr_9 \wedge \Delta Nr_{11} \wedge \Delta Nr_{12}$	f_3
$\Delta Nr_2 \wedge \Delta Nr_6$	$f_{4-\beta 1} \vee f_{(7-8)-PA1}$
$\Delta Nr_3 \wedge \Delta Nr_7$	$f_{4-\beta 2} \vee f_{(7-8)-PA2}$
$\Delta Nr_4 \wedge \Delta Nr_8$	$f_{4-\beta 3} \vee f_{(7-8)-PA3}$
ΔNr_1	$f_3 \vee f_5 \vee f_9$
$\Delta Nr_1 \wedge \Delta Nr_5 \wedge \Delta Nr_9 \wedge \Delta Nr_{10} \wedge \Delta Nr_{11}$	$f_3 \vee f_9$

Therefore, FDI approach to fault isolation follows a column view approach. From Table (3.6) it can be seen that the faults in each one of the blade root moment sensors is isolable ($f_{1-M,B,1,m}, f_{1-M,B,2,m} and f_{1-M,B,3,m}$).

Fault 3 in the generator speed sensor and Fault 9 a torque offset are also isolable.

To isolate a fault between the pitch sensor f_4 or actuator f_7 and f_8, of a pitch subsystem is not possible with the signature matrix (3.5), however it is remarkable to mention that a fault in an specific pitch subsystem can be isolated. For example if r_2 and r_6 activates, with the information provided by (3.6), it is possible to declare that a fault has occurred in the pitch subsystem of blade 1. What is not possible to determine is if the fault comes from the actuator or the sensor of the mentioned pitch subsystem. It happens the same with the pitch subsystems of blade 2 and 3. This means that it is possible to locate the pitch subsystem in which the fault occurs (pitch subsystem of blade 1, 2 or 3) even though that is not possible to determine if the fault is located in the actuator or the sensor.

As seen in this section the FDI approach to fault isolation follows a column view approach.

3.6.2 DX fault isolation

Fault detection can be considered as equivalent using ARRs in FDI and potential conflicts in DX [7], however fault isolation can be tackled in two different ways within the DX community: consistency-based, CBD, or abduction-based [14]. While consistency-based diagnosis tries to reject those behavioral modes which are not consistent with current observations, abduction-based diagnosis tries to explain current observation with a consistent behavioral mode assignment. In such sense, abduction-based diagnosis is closer to FDI approach than consistency-based. Moreover, for real complex dynamic systems there is no direct translation from the static consistency test to the dynamic one [6].

Based on the proposed framework by Cordier [7], the fault signature matrix is interpreted in DX CBD approach to fault isolation considering separately each line corresponding to a violated ARR, isolating R-conflicts (i.e., a set of components that are to be considered abnormal in order to be consistent with the observed fault signature) before searching for a common explanation, i.e., follows a row view of the table [43].

Row reasoning only looks for the activated fault signals and the corresponding rows in the fault signature matrix, applying this approach to (3.5), Table (3.7) is obtained:

Table 3.7: Fault Isolation DX approach

Logical Test	Diagnostic
ΔNr_1	$f_3 \vee f_5 \vee f_9$
ΔNr_2	$f_{1-M,B,1,m} \vee f_{4-\beta 1} \vee f_{7-PA1} \vee f_{8-PA1}$
ΔNr_3	$f_{1-M,B,2,m} \vee f_{4-\beta 2} \vee f_{7-PA2} \vee f_{8-PA2}$
ΔNr_4	$f_{1-M,B,3,m} \vee f_{4-\beta 3} \vee f_{7-PA3} \vee f_{8-PA3}$
ΔNr_5	f_9
ΔNr_6	$f_{4-\beta 1} \vee f_{7-PA1} \vee f_{8-PA1}$
ΔNr_7	$f_{4-\beta 2} \vee f_{7-PA2} \vee f_{8-PA2}$
ΔNr_8	$f_{4-\beta 3} \vee f_{7-PA3} \vee f_{8-PA3}$
ΔNr_9	$f_3 \vee f_9$
ΔNr_{10}	f_9
ΔNr_{11}	$f_3 \vee f_9$
ΔNr_{12}	f_3

3.6.3 FDI and DX common limitations

When applying DX and FDI approaches to dynamic systems, since they may exhibit symptoms with different dynamics (fast, slow, ...), the use of binary codification of the result of the fault detection test associated to a given ARR (or potential conflict), some information has been lost. In particular, among others:

- The sign of the symptom is lost.

- The sensitivity of the symptom respect to each fault is lost.

- The order of system appearance of a given ARR with respect to the others is not recorded, since it is a static test.

- The persistency of the symptom indicator of each ARR after fault is independent of the others.

- The time to wait the appearance of a set of symptoms times to isolate the faults when it exists different appearance times is not considered.

- Instability of the fault detection test indicator (chattering) since the presence of noise and the binary test used.

In [43] opinion, the diagnosis provided by each approach could be improved if that information was considered.

Chapter 4

Calibration, Parameters Adjustment and System Identification

The motivation of this chapter is to tune analytical redundant relations (ARRs) obtained in Chapter 3, using the data and behavior of the systems variables observed in FAST simulations. To perform fault detection it is necessary to parameterize the analytical redundant relations and this can be done using real data or a simulator which in the case of this thesis is FAST simulator, which will be explained later in this chapter.

To parameterize the ARRs, system identification techniques are applied based on the data obtained by simulations performed in FAST and having as reference the physical and theoretical models introduced in Chapter 2 for the structures of the identified models. A parameters comparison between those obtained by the estimated models and the nominal ones used by FAST wind turbine reference is also done.

4.1   FAST Simulator

FAST [20], is an aeroelastic wind turbine simulator designed by the U.S. National Renewable Energy Laboratory (NREL) National Wind Technology Center and widely used for studying wind turbine control systems. Several FAST models of real and composite wind turbines of varying sizes are available in the public domain. In the case studied in this thesis, the NREL 5 MW baseline turbine made up of composite pieces representative of a real utility-scale turbine was used.

This three-bladed, variable speed turbine with full span blade pitch control is available in both onshore and offshore versions, including four variations of off-shore structures. The wind speed is considered as an external input generated by Fast Simulator. FAST can utilize either uniform or full-field turbulent wind input files, with the turbulent files generated by the NREL software TurbSim. TurbSim generates turbulence using one of several atmospheric turbulence models and implements the wind field by creating the requested number of grids of 3-dimensional wind turbine velocity components, where the grids march forward toward the turbine in time based on the average wind speed. In the wind input files different wind speeds can be generated. The constant wind inputs files are generated by NREL software IECWind, which is the type of wind used in the simulations performed in this master thesis.

4.1.1   Control Strategy

The controller designed and implemented in benchmark [32], operates as indicated in [37]. It operates in principle in four zones, Zone 1 is start up of the turbine, Zone 2 is power optimization, Zone 3 is constant power production, Zone 4 is high wind speed see e.g. [24]. It should be noticed that these control zones often are divided into more zones for implemental reasons, in order to handle the transitions between the control modes as smoothly as possible.

In Figure 4.1 the power curve for the wind turbine is plotted. From this figure it can be seen that, for wind speeds between 4 and 12 m/s, the turbine is controlled to obtain optimal power production. From winds greater than 12 m/s the control strategy changes to Zone 3 in which the objective is to track a constant

power reference. In this thesis all the analysis and tests are performed with the controller operating in Zone 3.
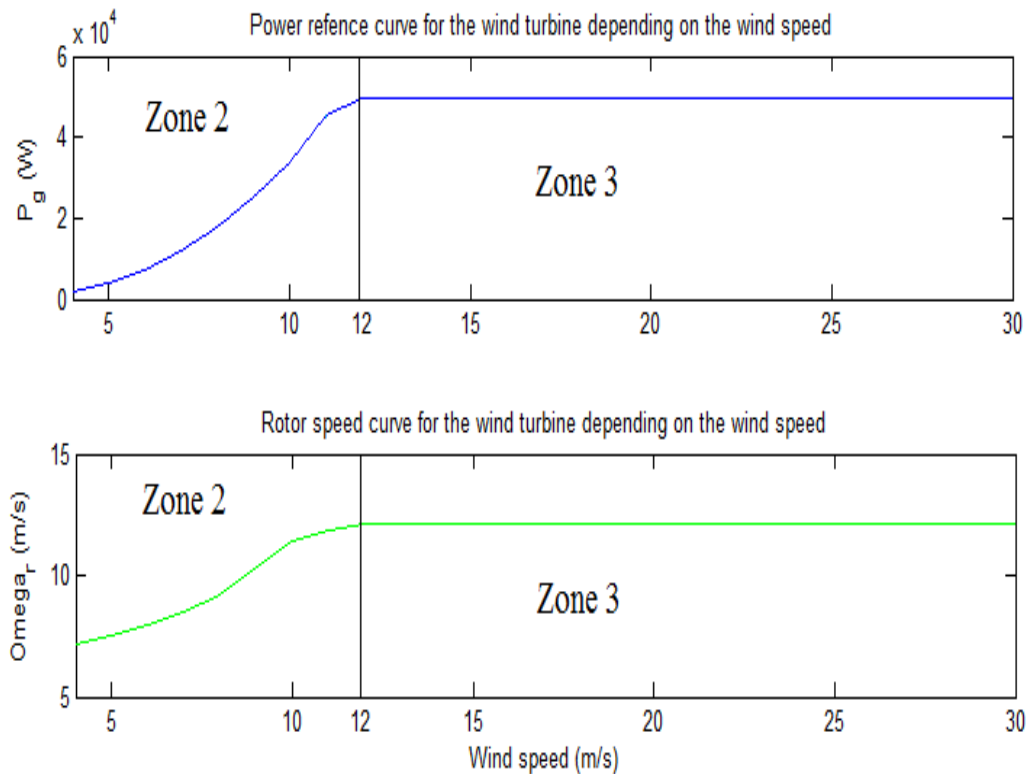

Figure 4.1: Power and Rotor speed curve depending on the wind speed

4.1.2 Parameters of NREL 5MW Wind Turbine Reference

The specifications of a representative utility-scale multimegawatt turbine now known as the NREL offshore 5-MW baseline wind turbine was developed by [21]. This wind turbine is a conventional three-bladed upwind variable-speed variable blade-pitch-to-feather-controlled turbine. To create the model, it was obtained some broad design information from the published documents of turbine manufacturers, with a heavy emphasis on the REpower 5M machine. Because detailed data was unavailable, however, they also used the publicly available properties from the conceptual models in the WindPACT, RECOFF, and DOWEC projects. A composite from these data was created, extracting the best available and most representative specifications. The report [21] documents the specifications of the NREL offshore 5-MW baseline wind turbineincluding the aerodynamic, structural, and control-system propertiesand the rationale behind its development. The model has been, and will likely continue to be, used as a reference by research teams throughout the world to standardize baseline offshore wind turbine specifications and to quantify the benefits of advanced land- and sea-based wind energy technologies.

The parameters of the report [21], used for characterizing the wind turbine subsystems in this thesis are shown below

4.1.3 Simulink Interface

As explained in [20], Simulink has the ability to incorporate custom Fortran routines in a block called an S-Function. The FAST subroutines have been linked with a MATLAB standard gateway subroutine in order

Table 4.1: Fault Scenarios

Parameter	Description	Value	Units
J_g	Generator inertia about the high speed shaft	kg m^2	534.116
B_{dt}	Drive train torsion damping coefficient	Nm/(rad/s)	6.125*10^6
N_g	Gear box ratio	97.0	-
J_r	Hub inertia about rotor axis	115.926*10^3	kg m^2
R	The distance from the rotor apex to the blade tip	63	m

to use the FAST equations of motion in an S-Function that can be incorporated in a Simulink model. This introduces tremendous flexibility in wind turbine controls implementation during simulation. Generator torque control, nacelle yaw control, and pitch control modules can be designed in the Simulink environment and simulated while making use of the complete nonlinear aeroelastic wind turbine equations of motion available in FAST. The wind turbine block, as shown in Figure **??**, contains the S-Function block with the FAST equations of motion. It also contains blocks that integrate the accelerations to get velocities and displacements. Thus the equations of motion are formulated in the FAST S-function but solved using one of the Simulink solvers.
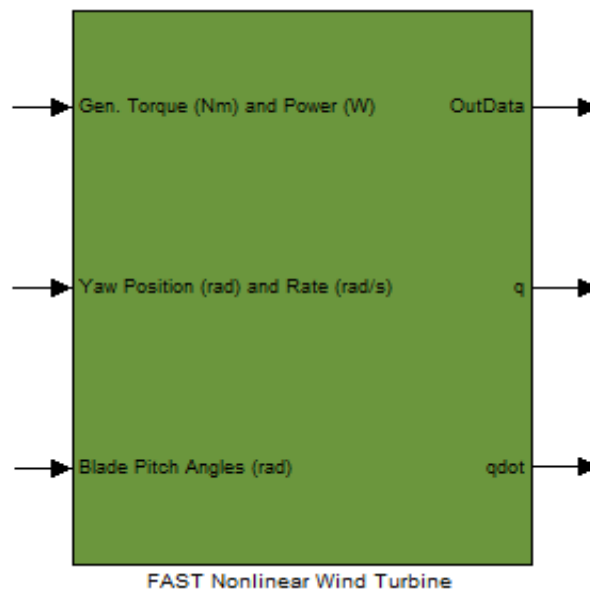

Figure 4.2: FAST Non Linear Wind Turbine

4.2 Selection of identification signals

The ARRs adjustment will be done using the tools of parameters identification. The first difficulty that appears is that the wind turbine operates in closed loop and cannot be operated in open loop. Because of that it is necessary to apply perturbation signals. When perturbation signals are applied to the system, all its relevant modes are excited and it is aimed to achieve a good signal-to-noise ratio between the output signal content due to the excitations and due to the disturbances and noise. From the identification point of view, it can be considered that applying the larger perturbation signals possible would be good to the identification process. However this is not the case in practice:

  - If a large perturbation signal is applied, the system will operate in a wide range and this would result in a violation of the linear condition to be violated.

- Large perturbations can lead the system to suffer fatigue and cause damages. Torque perturbations are reflected in the output generator power, excites the drivetrain and it influences the generator speed. In the case of pitch perturbations, these cause variations in the rotor speed and therefore in the aerodynamic torque.

- The perturbation signals must be designed in order to satisfy the actuator limits constraints.

Considering the system showed in picture 4.3, the controlled inputs entering the wind turbine are the pitch angle $\beta(t)$ and the generator torque $T_g(t)$. The wind speed $v_r(t)$ may be considered a disturbance in the context of system identification and it is modeled as a PRBS signal using the FAST preprocessor IECWind with that purpose. Using these three input signals we excite the drive train dynamics which are the ones of interest in this case. It is important to ensure not only that the turbine operates in a fairly narrow operating range (in this case the wind turbine operates at a constant wind speed), but also that the controllers act as an LTI system (the controllers of the turbines are linear controllers as it can be seen in FAST-Simulink interface).
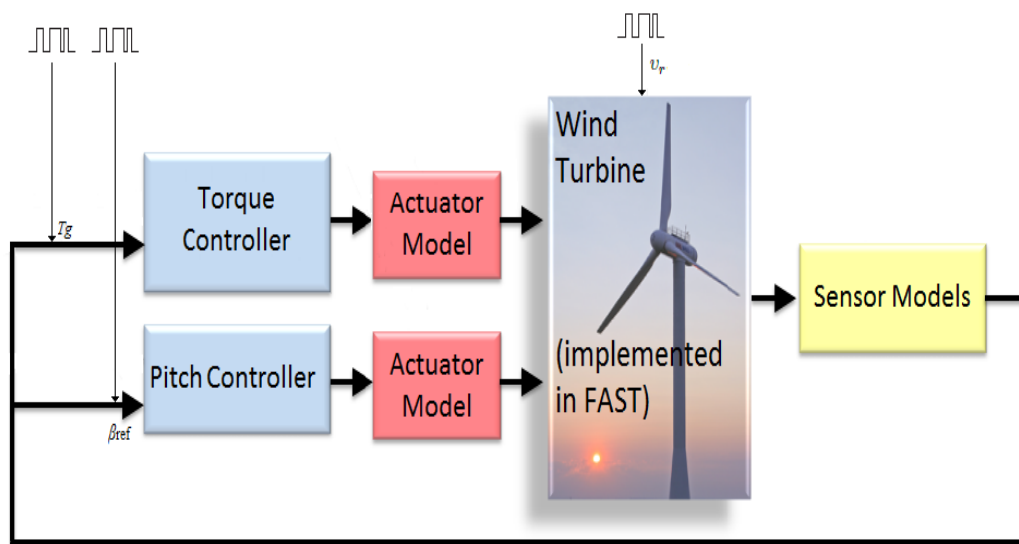
Figure 4.3: Wind Turbine System with perturbations introduced.

The following pictures shows the pitch, torque and wind perturbation signals designed for the drive train subsystem identification of the NREL offshore 5-MW baseline wind turbine implemented in FAST simulator. The behavior of the turbine was simulated at a constant wind speed of 24 m/s and the perturbation signals were applied. The signals were a pseudo-random binary signal of ±3500 (Nm) for the torque perturbation and a pseudo-random binary signal of ±3 (deg).

Figure 4.4: Pitch perturbation signal

Figure 4.5: Torque perturbation signal

Figure 4.6: PRBS signal designed for the wind speed using FAST-IECWind preprocessor.

4.3 Generator and Converter Subsystem
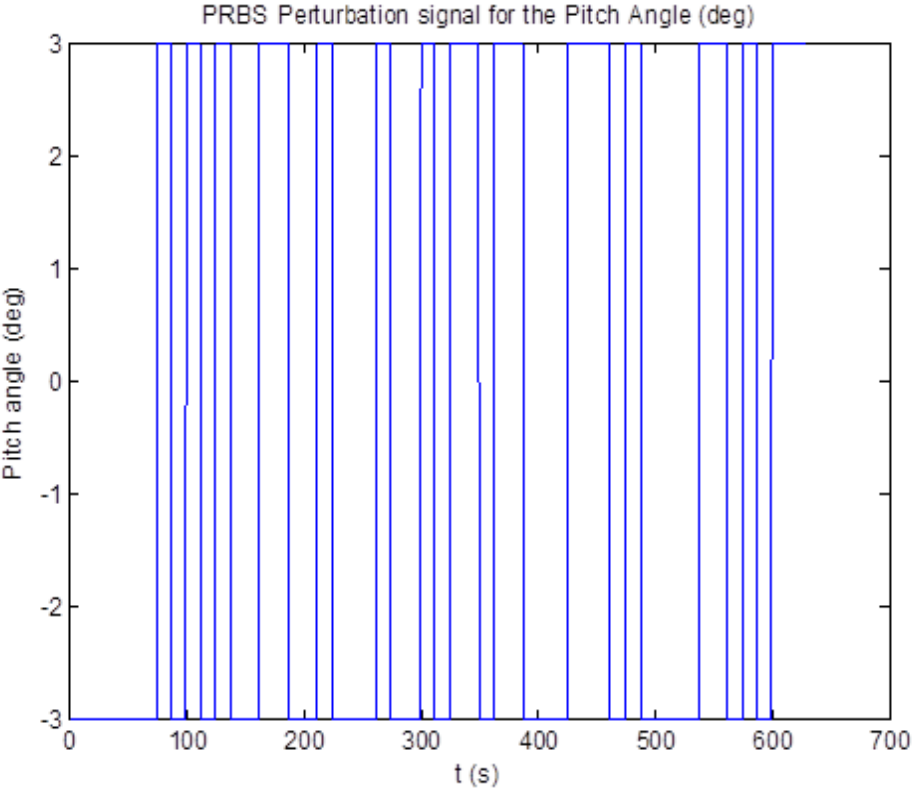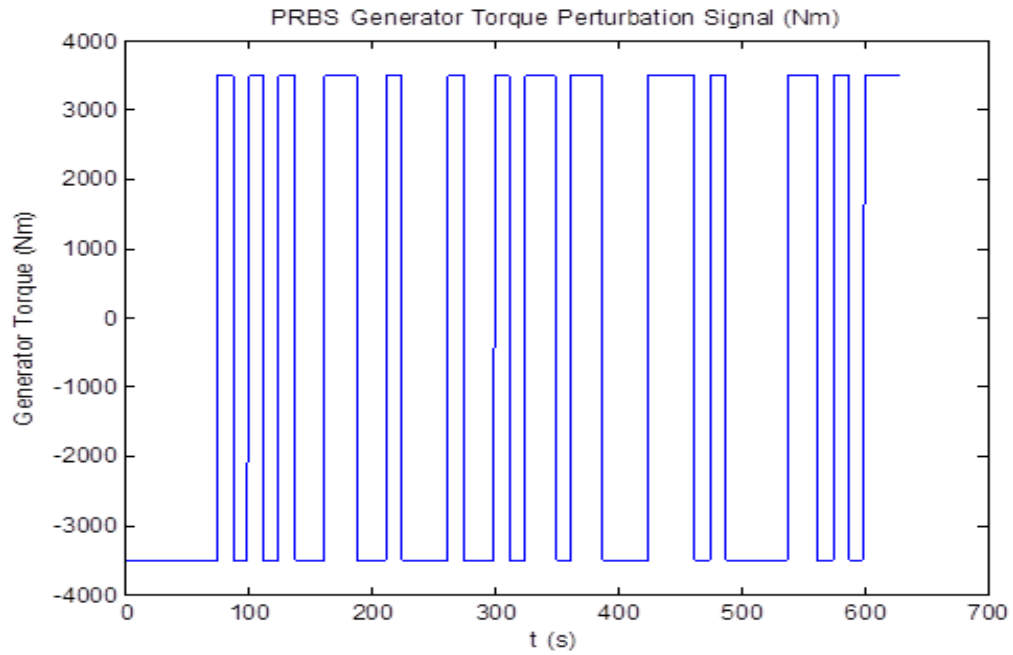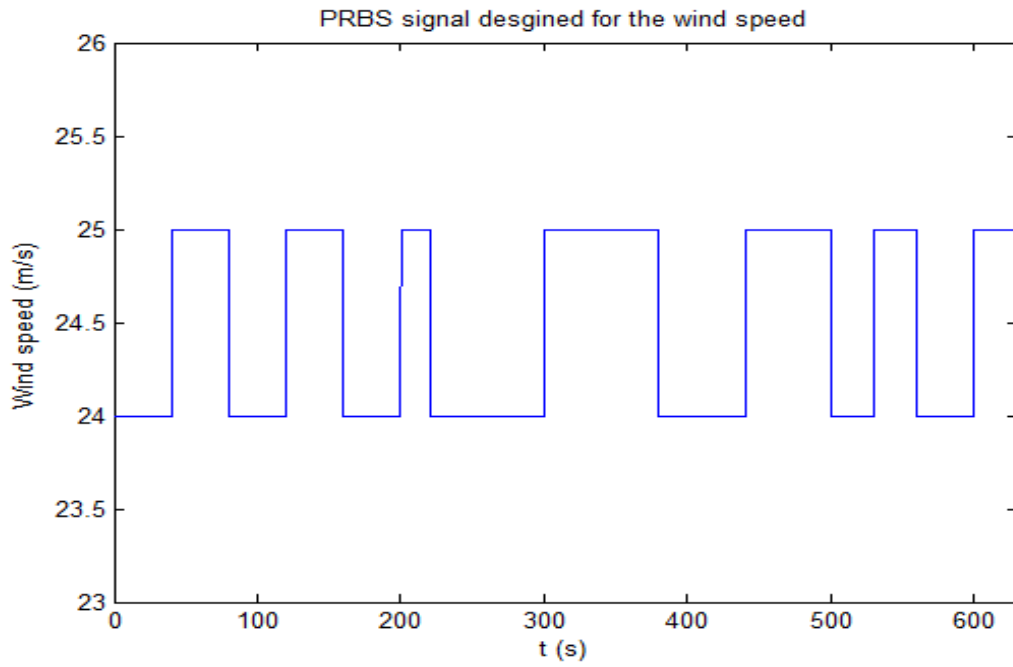
The model for the generator and converter subsystem proposed in equation (2.112) suggest that we can use a first order discrete ARX structure for the identified model. This is detailed below.

Equation (2.112) can be rewritten as:

$$\tau_g s T_g(s) + T_g(s) = T_{gref}(s) \tag{4.1}$$

Using the transformation $s = \frac{z-1}{T_s}$ to obtain the model in discrete time with T_s is the sampling time we obtain the following:

$$\tau_g \left(\frac{z-1}{T_s} \right) T_g(z) + T_g(z) = T_{gref}(z) \tag{4.2}$$

Multiplying by the factor z^{-1} and grouping the terms, the following discrete-time model can be obtained which suggests the structure to be used in the identification using data.

$$\frac{\tau_g}{T_s} T_g(z) + \left(1 - \frac{\tau_g}{T_s} \right) z^{-1} T_g(z) = z^{-1} T_{gref}(z) \tag{4.3}$$

Using FAST simulator with a constant wind input of 24 m/s applied to the wind turbine system, the data to be used for the identification process is obtained. The 60 percent of the samples are used for estimating the model and the other 40 percent is used for validation purposes.

The identified model has as an input the torque reference signal $T_{gref}(t)$ with the perturbation showed in Figure 4.5. The output of the identified system is the measured generator torque $T_g(t)$.

The estimated discrete-time ARX model $A(z)y(t) = B(z)u(t) + e(t)$ is presented below.

$$\begin{aligned} A(z) &= 1 - 0.536(\pm 9.335 \times 10^{-9})z^{-1} \\ B(z) &= 0.000464(\pm 9.327 \times 10^{-12})z^{-1} \end{aligned} \tag{4.4}$$

In Figure 4.7 it is observed a fit of 99.98"%" of fit between the measured and simulated output of the ARX model.

Rewriting equation (4.3) in order to compare the parameters with the ARX estimated model (4.4), the following equation is obtained.

$$T_g(z) + \left(\frac{T_s}{\tau_g} - 1 \right) z^{-1} T_g(z) = \frac{T_s}{\tau_g} z^{-1} T_{gref}(z) \tag{4.5}$$

4.3.1 Comparison with FAST implemented Generator Model

The generator and converter model presented in equation (2.112) is the one implemented in FAST simulator within the benchmark model [32]. In this section a comparison between the values of the generator model parameters used in FAST with the ones estimated during the identification process showed on (4.6) is done.

From the data obtained during FAST simulations, it is observed that there exists a gain of $K = 10^{-3}$ in the generator torque output of FAST nonlinear plant respect to the generator torque output of the model of the actuator.

Therefore to compare both models, the ARX discrete time identified model (4.4) and the generator model implemented in FAST simulator showed in equation (2.112), the value of this gain has to be taken into account.

$$T_g(z) + \left(\frac{T_s}{\tau_g} - 1 \right) z^{-1} T_g(z) = K \frac{T_s}{\tau_g} z^{-1} T_{gref}(z) \tag{4.6}$$

After comparing (4.6) and (4.4), we obtain the following parameters values.

$$\frac{T_s}{\tau_g} - 1 = 0.536 \Rightarrow \tau_g = 0.027 \tag{4.7}$$

The value of the estimated parameter of the ARX discrete system (4.7) is almost the same as the nominal value of the parameter used in FAST and proposed in the benchmark model [32]: $\tau_g = {}^1\!/_{50} = 0.02$, which is the same model taken by reference in this work ((2.112)).
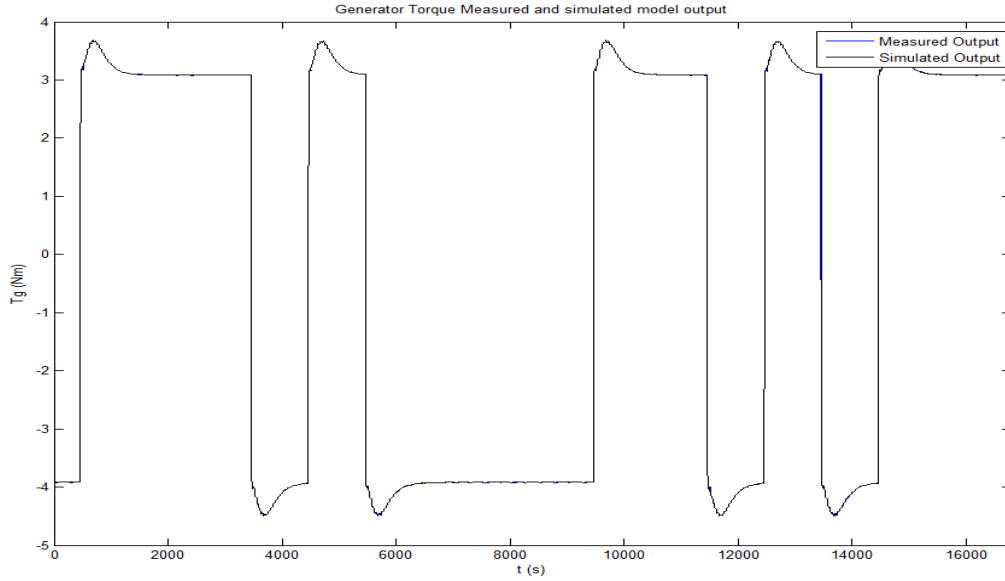

Figure 4.7: Measured and Simulated Output of the Generator/Converter ARX model

4.4   Pitch Subsystem

In this section the objective is to obtain the discrete model of the pitch subsystem presented in (2.60) in order to propose the structure of the model to be used in the identification process. Once this discrete model is obtained, the parameters values can be estimated and compared to the nominal ones obtained in FAST.

Using the transformation $s = \frac{z-1}{T_s}$, the following second order discrete function is deduced.

$$\frac{\beta(z)}{\beta_{ref}(z)} = \frac{\omega_n^2}{\left(\frac{z-1}{T_s}\right)^2 + 2\zeta\omega_n\left(\frac{z-1}{T_s}\right) + \omega_n^2} \tag{4.8}$$

System (4.8) can be written as:

$$\beta(z)\left(\frac{z-1}{T_s}\right)^2 + 2\zeta\omega_n\left(\frac{z-1}{T_s}\right) + \omega_n^2 = \beta_{ref}(z)\omega_n^2 \tag{4.9}$$

Multiplying equation (4.9) by z^{-2} and grouping terms equation (4.10) is obtained.

$$z^{-2}\beta(z)\left(1 - 2\zeta\omega_n T_s + T_s^2\omega_n^2\right) + z^{-1}\beta(z)\left(-2 + 2\zeta\omega_n T_s\right) \\ +\beta(z) = T_s^2\omega_n^2 z^{-2}\beta_{ref}(z) \tag{4.10}$$

Based on the discrete model (4.10), a second order discrete-time ARX function is proposed to identify the pitch subsystem. Using the data obtained during a FAST simulation with the constant wind speed of 24 m/s the discrete time ARX model showed on (4.11) is obtained.

$$A(z) = 1 - 1.829(\pm 2.852 \times 10^{-13})z^{-1} + 0.8465(\pm 2.515 \times 10^{-13})z^{-2} \\ B(z) = 0.009114(\pm 1.505 \times 10^{-13})z^{-1} + 0.008622(\pm 1.768 \times 10^{-13})z^{-2} \tag{4.11}$$

In figure 4.8 it is plotted the measured (FAST Data) and simulated (ARX model) output with a fit of 100 "%".

Comparing the polynomial A(z) of (4.11) and the left side of the equation (4.10), the following system of equations to determine the parameters of the pitch subsystem is deduced.

$$\begin{cases} 1 - 2\zeta\omega_n T_s + T_s^2\omega_n^2 = 0.8465 \\ -2 + 2\zeta\omega_n T_s = -1.829 \end{cases} \tag{4.12}$$

From the system of equations (4.12), we have the following values for the parameters ζ and ω_n

$$\begin{aligned} \zeta &= 0.6157 \\ \omega_n &= 10.5830 \end{aligned} \tag{4.13}$$

The parameters showed in (4.13) are very similar to the nominal ones for ζ and ω_n presented in equation (5.1) used in benchmark [32] and also in FAST simulator.
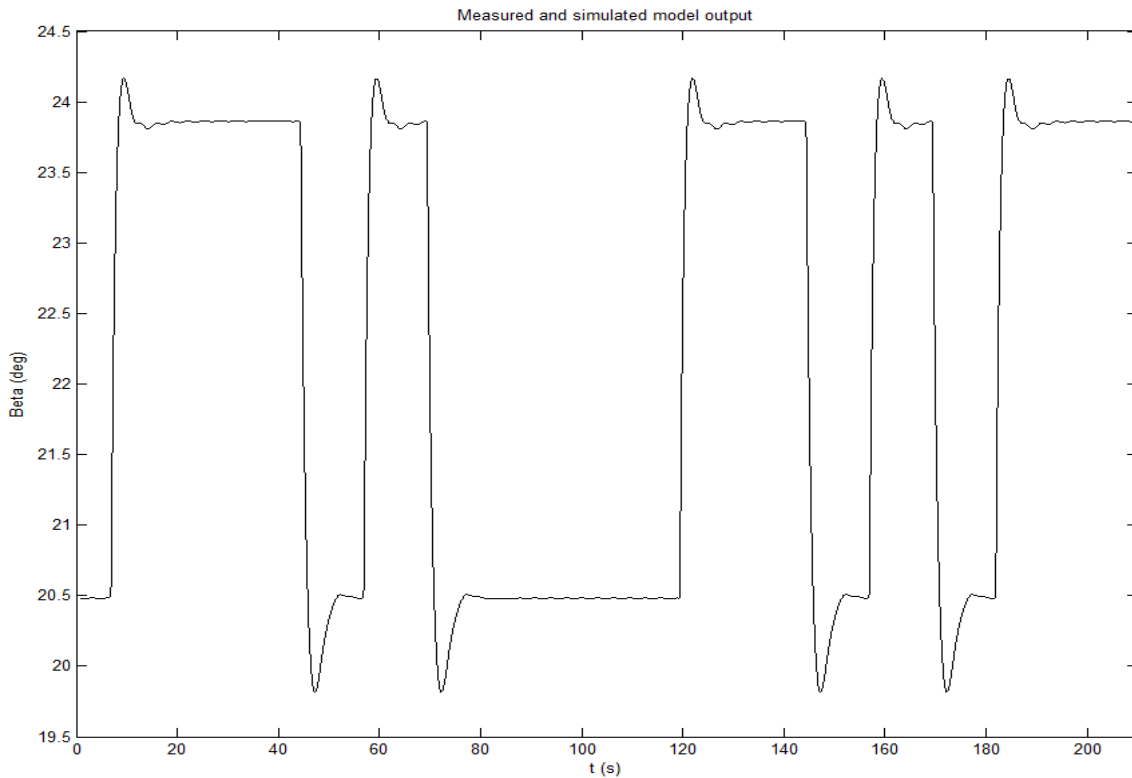
Figure 4.8: Measured and Simulated compared output of the pitch ARX model

4.5 Power Subsystem

In this section we adjust the parameter η_g that relates the generated power in the wind turbine with the generator torque and speed. During FAST simulation using a constant wind speed of 24 m/s, it is observed that the model of the generated power (2.113) corresponds to the generator power data obtained from the simulations, therefore this static model it is validated.

In the Figures 4.9, ?? and ?? the values for the generator power and the generator speed and torque in steady state are visualized. It is observed that the values obtained, validate the model proposed in (2.113)).

Calculating the generator efficiency from the observed data it can be seen that it corresponds to the nominal generator efficiency used in FAST.

$$\frac{P_g(t)}{\omega_g(t)\tau_g(t)} = \eta_g \rightarrow \eta_g = \frac{49570}{1174 \times 43.09} = 0.98 \tag{4.14}$$

Figure 4.9: Generator speed, generator torque and power in steady state

4.6 Blade Root Moment Dynamics

The Blade Root Moment dynamics obtained using the Fast Simulator shows an oscillating behavior in steady state regime. This is illustrated in the following figures.

In Figure 4.10 it is observed the Blade Root Moments different phase shifts in steady state regime.
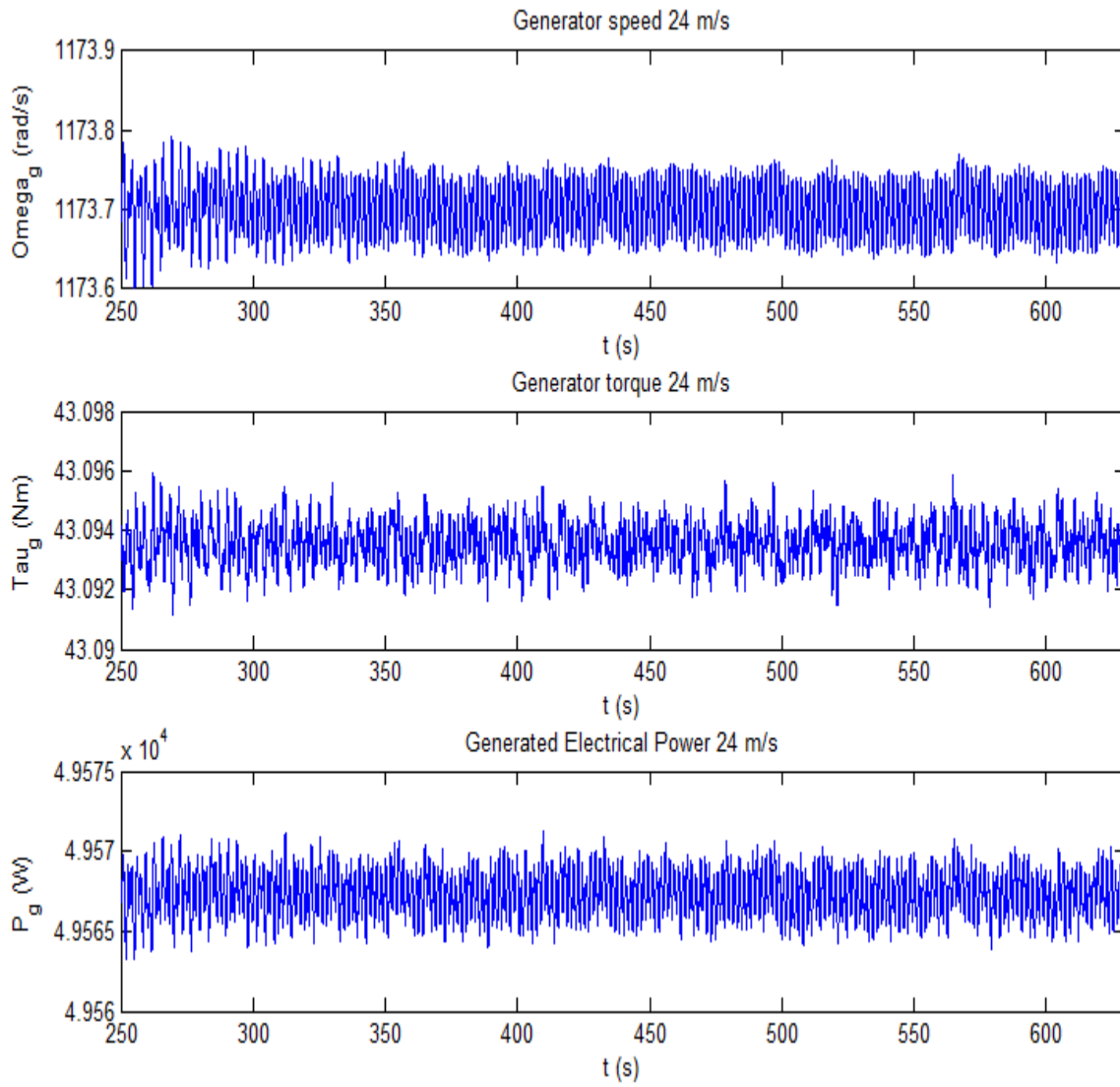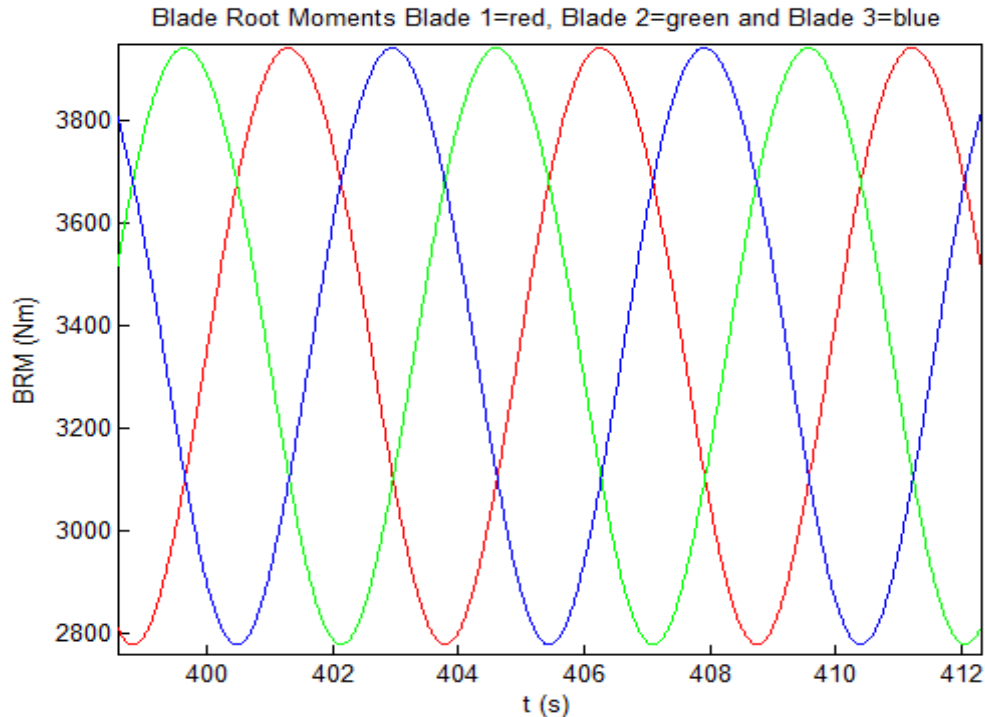

Figure 4.10: Blade Root Moment phase shifts in steady state regime

Several tests with different wind speed inputs were made. It was observed that there is a different phase shift in each of the three blade root moments in steady state behavior when applying different wind inputs. This is illustrated in Table 4.7.

Along the differences in the phase shifts, it was also observed a variable signal amplitude when applying a zoom on the upper peaks of the signal. The steady state behavior of the blade root moments can be seen in Figure 4.11.

Because of the variable phase shifts and amplitudes present among the blade root moments for different wind inputs, the estimated models were not so accurate approximating the behavior of the blade root moments dynamics. After observing these results another approach was proposed as explained in the next section.

4.6.1 Mean Blade Root Moment Model

Zone 2. Wind speeds < 12 m/s

In this zone there is no pitch control, therefore the obtained model only depends of the wind speed. In Table 4.2 are listed the blade root moment values obtained for each wind input.

Zone 3. Wind Speeds > 12 m/s. In this second zone there is pitch control. Therefore we only estimate the parameter of the pitch angle which depends on the wind speed.

After obtaining the measurements of the mean pitch angles and mean blade root moments for each of the blades the following models were proposed to approximate the blade root moments dynamics.

For wind speeds < 12 m/s.

$$\bar{M}_{B,i(t)} = a_1 \bar{v}(t) + a_0 \tag{4.15}$$
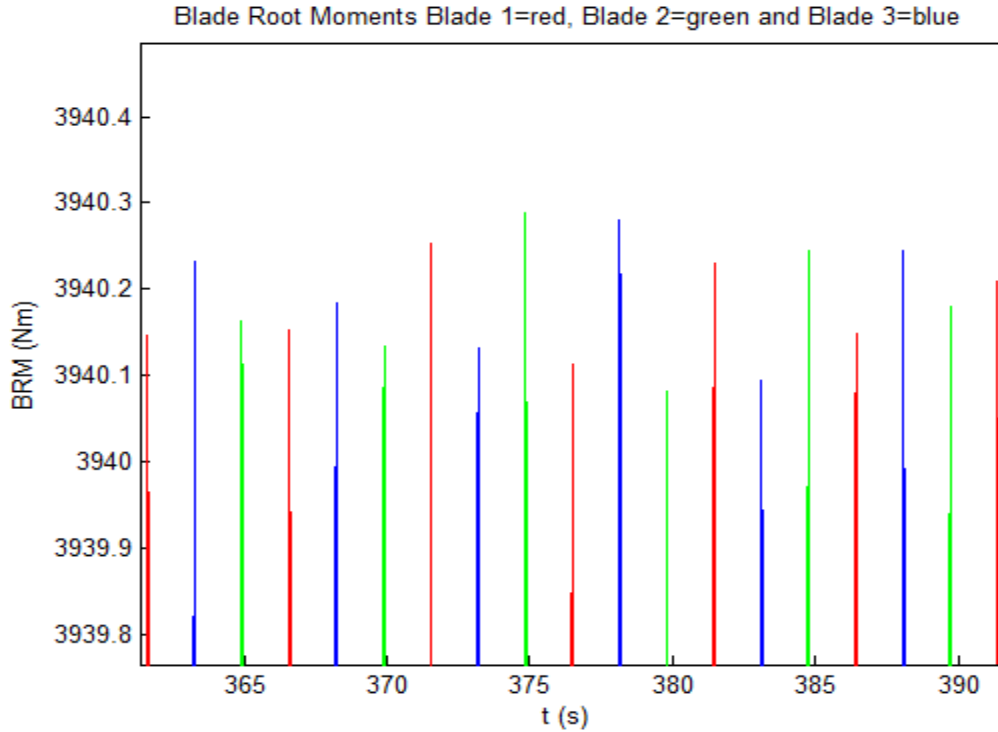
Figure 4.11: Blade Root Moments different amplitudes in steady state regime

$$\bar{M}_{B,i(t)} = a_2\,\bar{v}(t)^2 + a_1\,\bar{v}(t) + a_0 \tag{4.16}$$

$$\bar{M}_{B,i(t)} = a_3\bar{v}(t)^3 + a_2\,\bar{v}(t)^2 + a_1\,\bar{v}(t) + a_0 \tag{4.17}$$

For wind speeds $\geq$ 12 m/s.

$$\bar{M}_{B,i(t)} = a_1\,\bar{\beta}_i(t) + a_0 \tag{4.18}$$

$$\bar{M}_{B,i(t)} = a_2\,\bar{\beta}_i(t)^2 + a_1\,\bar{\beta}_i(t) + a_0 \tag{4.19}$$

$$\bar{M}_{B,i(t)} = a_3\bar{\beta}_i(t)^3 + a_2\,\bar{\beta}_i(t)^2 + a_1\,\bar{\beta}_i(t) + a_0 \tag{4.20}$$

The absolute, mean and quadratic errors were calculated for each one of the estimated models and their values are detailed in Table 4.5 for the models of Zone 1 and Table 4.6 for the models of Zone 2.

For the models of zone 1 which are functions of the wind speed, the error of the third order model is minor than the one of second order. This last one will be chosen as the blade root moment estimated mean model because there is not too much of a difference in the magnitude of the error compared to the third order one and it's complexity is smaller.

The results of the estimated models were very similar for the blade root moments of each blade and are shown in Figures 4.12 and 4.13.
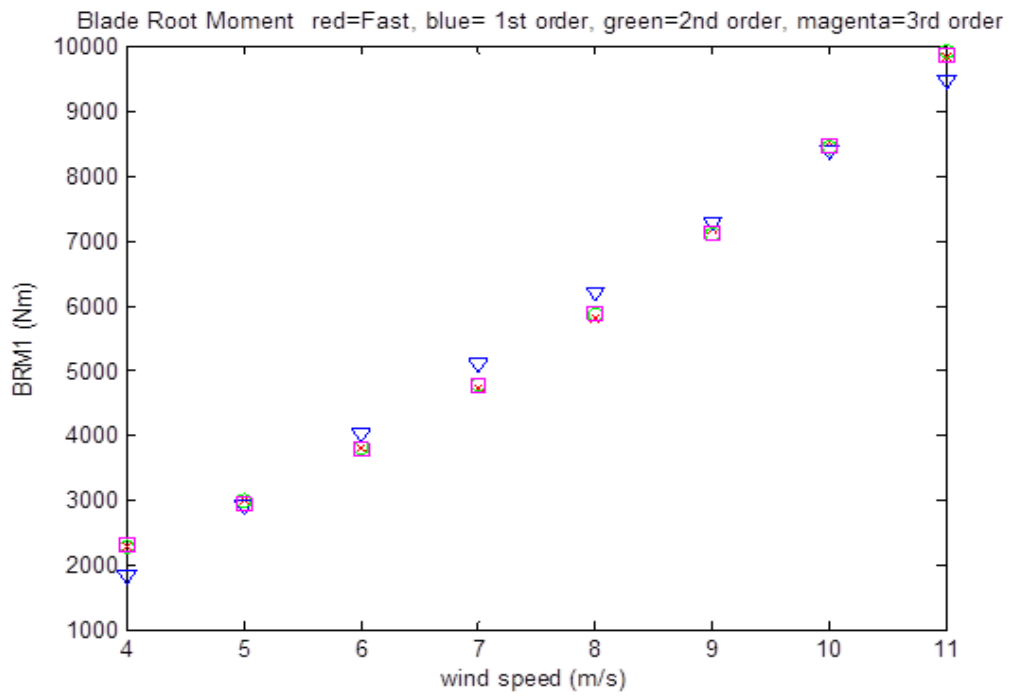
Figure 4.12: Blade Root Moments estimated models for control zone 1. Wind speeds < 12 m/s
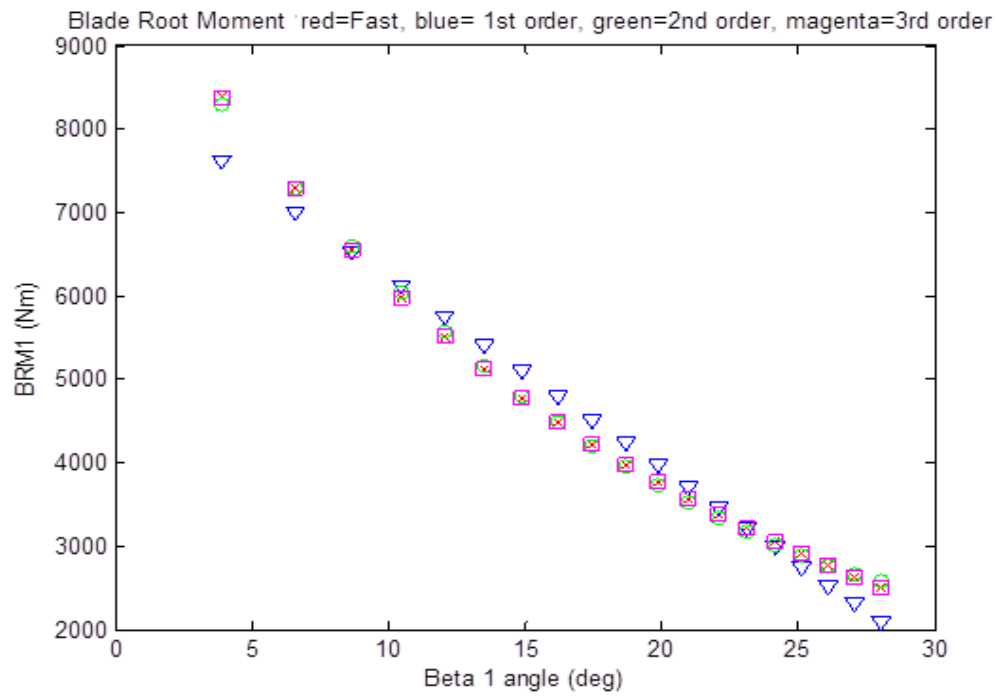
Figure 4.13: Blade Root Moments estimated models for control zone 2. Wind speeds ≥ 12 m/s

Table 4.2: Blade Root Moments obtained for each wind

Wind Speed (m/s)	BRM Mean
4	
BRM1	2.2865e+03
BRM2	2.2875e+03
BRM3	2.2866e+03
5	
BRM1	2.9798e+03
BRM2	2.9796e+03
BRM3	2.9866e+03
6	
BRM1	3.8016e+03
BRM2	3.8040e+03
BRM3	3.7962e+03
7	
BRM1	4.7350e+03
BRM2	4.7319e+03
BRM3	4.7285e+03
8	
BRM1	5.8085e+03
BRM2	5.8102e+03
BRM3	5.8067e+03
9	
BRM1	7.1670e+03
BRM2	7.1690e+03
BRM3	7.1743e+03
10	
BRM1	8.5128e+03
BRM2	8.5153e+03
BRM3	8.5084e+03
11	
BRM1	9.8358e+03
BRM2	9.8365e+03
BRM3	9.8401e+03

Table 4.3: Mean values of the pitch angles and the wind speed

Wind Speed (m/s)	BRM Mean	Mean Beta
12		
BRM1	8.3884e+03	
BRM2	8.3877e+03	3.9064
BRM3	8.3926e+03	
13		
BRM1	7.2871e+03	
BRM2	7.2885e+03	6.6071
BRM3	7.2930e+03	
14		
BRM1	6.5493e+03	
BRM2	6.5526e+03	8.6733
BRM3	6.5563e+03	
15		
BRM1	5.9815e+03	
BRM2	5.9864e+03	10.4503
BRM3	5.9889e+03	
16		
BRM1	5.5167e+03	
BRM2	5.5231e+03	12.0524
BRM3	5.5240e+03	
17		
BRM1	5.1233e+03	
BRM2	5.1308e+03	13.5320
BRM3	5.1301e+03	
18		
BRM1	4.4839e+03	
BRM2	4.7911e+03	14.9157
BRM3	4.7888e+03	
19		
BRM1	4.4839e+03	
BRM2	4.4923e+03	16.2203
BRM3	4.4885e+03	
20		
BRM1	4.2183e+03	
BRM2	4.2266e+03	17.4687
BRM3	4.2213e+03	
21		
BRM1	3.9806e+03	
BRM2	3.9884e+03	18.6767
BRM3	3.9818e+03	
22		
BRM1	3.7646e+03	
BRM2	3.7714e+03	19.8444
BRM3	3.7639e+03	

Table 4.4: Mean values of the pitch angles and the wind speed

Wind Speed (m/s)	BRM Mean	Mean Beta
23		
BRM1	3.5652e+03	
BRM2	3.5707e+03	20.9696
BRM3	3.5628e+03	
24		
BRM1	3.3805e+03	
BRM2	3.3842e+03	22.0561
BRM3	3.3764e+03	
25		
BRM1	3.2079e+03	
BRM2	3.2110e+03	23.1117
BRM3	3.2037e+03	
26		
BRM1	3.0497e+03	
BRM2	3.0500e+03	24.1405
BRM3	3.0436e+03	
27		
BRM1	2.9012e+03	
BRM2	2.9000e+03	25.1451
BRM3	2.8947e+03	
28		
BRM1	2.7623e+03	
BRM2	2.7597e+03	26.1265
BRM3	2.7560e+03	
29		
BRM1	2.6321e+03	
BRM2	2.6285e+03	27.0848
BRM3	2.6264e+03	
30		
BRM1	2.5096e+03	
BRM2	2.5055e+03	28.0216
BRM3	2.5052e+03	

Table 4.5: Zone 1. Error values obtained for each one of the estimated BRMs model

	First Order	Second Order	Third Order
Absolute Error			
BRM1	2102.8	364.8048	306.1333
BRM2	2103.8	368.5476	306.5143
BRM3	2116	379.8476	309.4190
Mean Error			
BRM1	262.8500	45.6006	38.2667
BRM2	262.9750	46.0685	39.1143
BRM3	264.5000	47.4810	39.7024
Mean Quadratic Error			
BRM1	5.7320	2.3875	2.1871
BRM2	5.7334	2.3997	2.2112
BRM3	5.7500	2.4362	2.2277

Table 4.6: Zone 2. Error values obtained for each one of the estimated BRMs model

	First Order	**Second Order**	**Third Order**
Absolute Error			
BRM1	4540.1	714.8686	46.4837
BRM2	4475.5	719.7803	57.9281
BRM3	4514.7	687.3887	58.2076
Mean Error			
BRM1	238.9503	37.6247	2.4465
BRM2	235.5528	37.8726	3.0488
BRM3	237.6163	36.1784	3.0636
Mean Quadratic Error			
BRM1	3.5463	1.4072	0.3588
BRM2	3.5210	1.4118	0.4006
BRM3	3.5364	1.3799	0.4015

Table 4.7: Phase Shifts between Blade Root Moments

Wind Speed (m/s)	**BRM3-BRM1 (rad)**	**BRM2-BRM1 (rad)**	**Period BRM1 (s)**
7	2.0358	4.1469	7.1
8	2.0145	4.0289	6.55
9	2.1166	4.1166	5.8
10	2.1749	4.2290	5.2
11	2.0944	4.0656	5.1
12	2.1363	4.1469	5
13	2.1363	4.2776	5
14	2.0517	4.2315	4.9
15	2.1363	4.1469	5
16	2.1363	4.2776	5
17	2.1363	4.1469	5
18	2.1363	4.1469	5
19	2.0106	4.1469	4.9
20	2.1363	4.1469	5
21	2.0517	4.2315	4.9
22	2.0517	4.2315	4.9
23	2.0106	4.1469	5
24	2.1363	4.1469	5
25	2.0106	4.1469	5
26	2.0106	4.1469	5
27	2.0106	4.1469	5

4.7 Drive Train Subsystem

The equations (2.47), (2.48) and (2.49) describe the drive train dynamics and the behavior of all this subsystems variables are analyzed in FAST simulator. In steady state the rotor and generator accelerations are zero.

Therefore equation (2.48) is the following:

$$0 = \frac{K_{dt}}{N_g}\theta_\Delta(t) + \frac{B_{dt}}{N_g}\omega_r(t) - \frac{B_{dt}}{N_g^2}\omega_g(t) - B_g\omega_g(t) - T_g(t) \qquad (4.21)$$

Multiplying (4.21) by $-Ng$ we obtain equation (4.22).

$$0 = -K_{dt}\theta_\Delta(t) - B_{dt}\omega_r(t) + \frac{B_{dt}}{N_g}\omega_g(t) + N_gB_g\omega_g(t) + N_gT_g(t) \qquad (4.22)$$

In steady state it is observed that relation (4.23), between the rotor speed and generator speed is satisfied

$$\omega_r(t) - \frac{1}{N_g}\omega_g(t) = 0 \qquad (4.23)$$

Therefore equation (4.24) is also satisfied

$$\theta_r(t) - \frac{1}{N_g}\theta_g(t) = K = \theta_\delta(t) \qquad (4.24)$$

Substituting equations (4.23) and (4.24) in eq. (4.21) the following equation is obtained:

$$T_g(t) = K - B_g\omega_g(t) \qquad (4.25)$$

Analytical redundant relation (3.17) can be written in the following way:

$$\omega_{g,m}(t) = -\frac{1}{B_g}T_{g,m}(t) + K \qquad (4.26)$$

From the simulations in FAST it was observed that the measured variables generator speed and generator torque were related in the way suggested in equation (4.26). These two variables are showed in Figure 4.14.

In Figure 4.14 can be seen that exists an inverse-proportional relation with a displacement between the variables generator torque and generator speed. This behavior is described by the analytical redundant relation (3.17), therefore the theoretical ARR corresponds to the generator speed and generator torque observed in FAST. From Figure 4.14, the parameters of ARR (4.26) can be estimated.

$$\begin{aligned} -\tfrac{1}{B_g} &= -27.2534 \Rightarrow B_g = 0.0367 \\ K &= 2348.1 \end{aligned} \qquad (4.27)$$

The comparison between the generator speed estimated with ARR (4.26) and the generator speed measured in FAST is showed in Figure 4.15. It can be observed that there is a very good fit between these two signals.

Analytical redundant relation (4.23) can be written in the following way:

$$\omega_g(s) = N_g\omega_r(s) \qquad (4.28)$$

In Figure 4.16 can be seen that exists a proportional relation between the generator and rotor speed. This behavior is described by the analytical redundant relation (4.28), therefore the theoretical ARR corresponds to the generator and rotor speed observed in FAST.

This proportionality between generator and rotor speed is the gearbox ratio parameter specificated in the modeling chapter. From Figure 4.16, the gearbox ratio parameter of ARR (4.28) can be estimated. The mean value of the gearbox ratio estimated from the rotor and generator speed data from FAST is $N_g = 97$, this estimated value of the gearbox ratio corresponds to the nominal value considered in the models.

The comparison between the generator speed estimated with ARR (4.28) and the generator speed measured in FAST is showed in Figure 4.17. It can be observed that fit between the estimated and the measured variable is excellent.
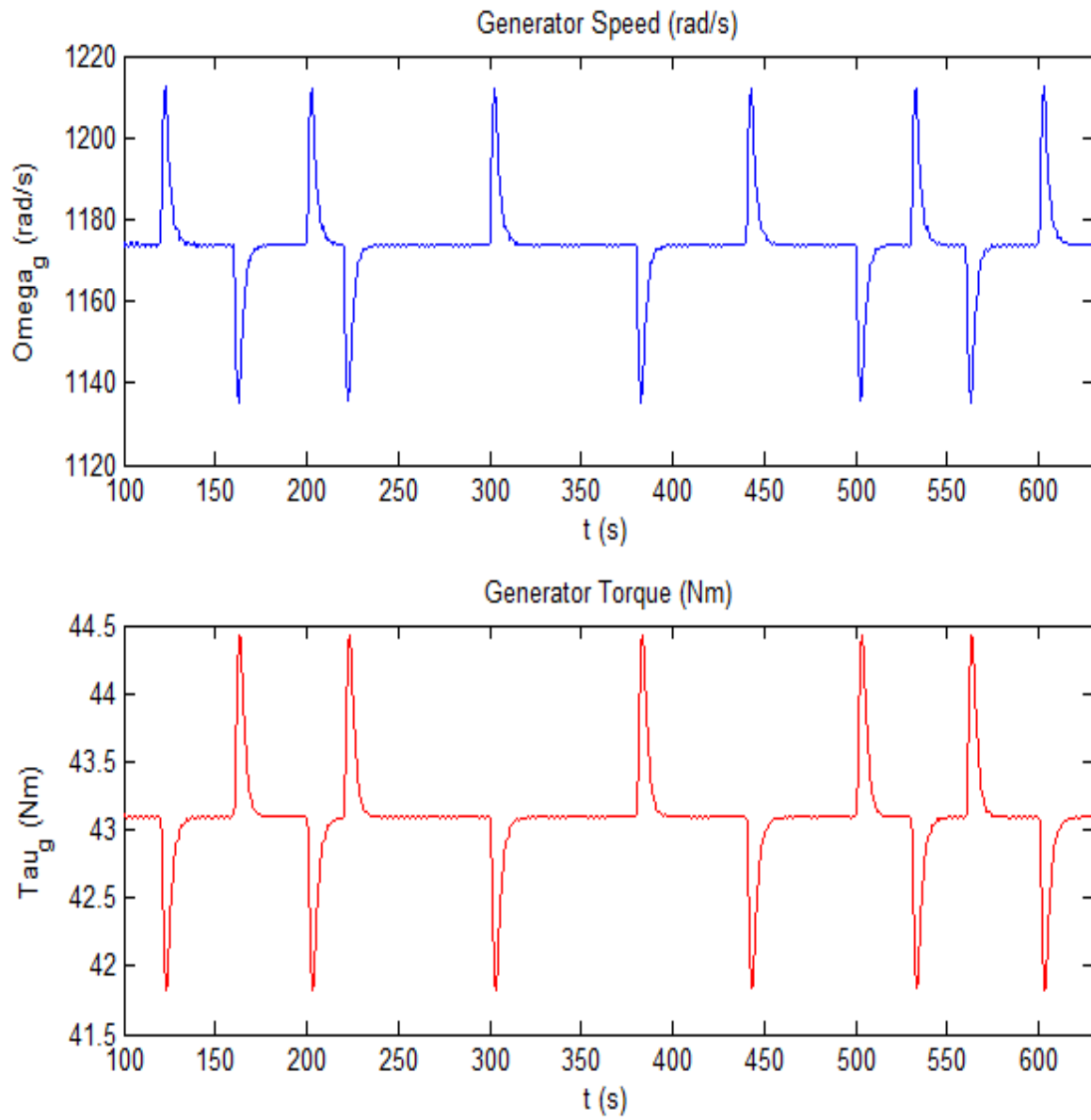
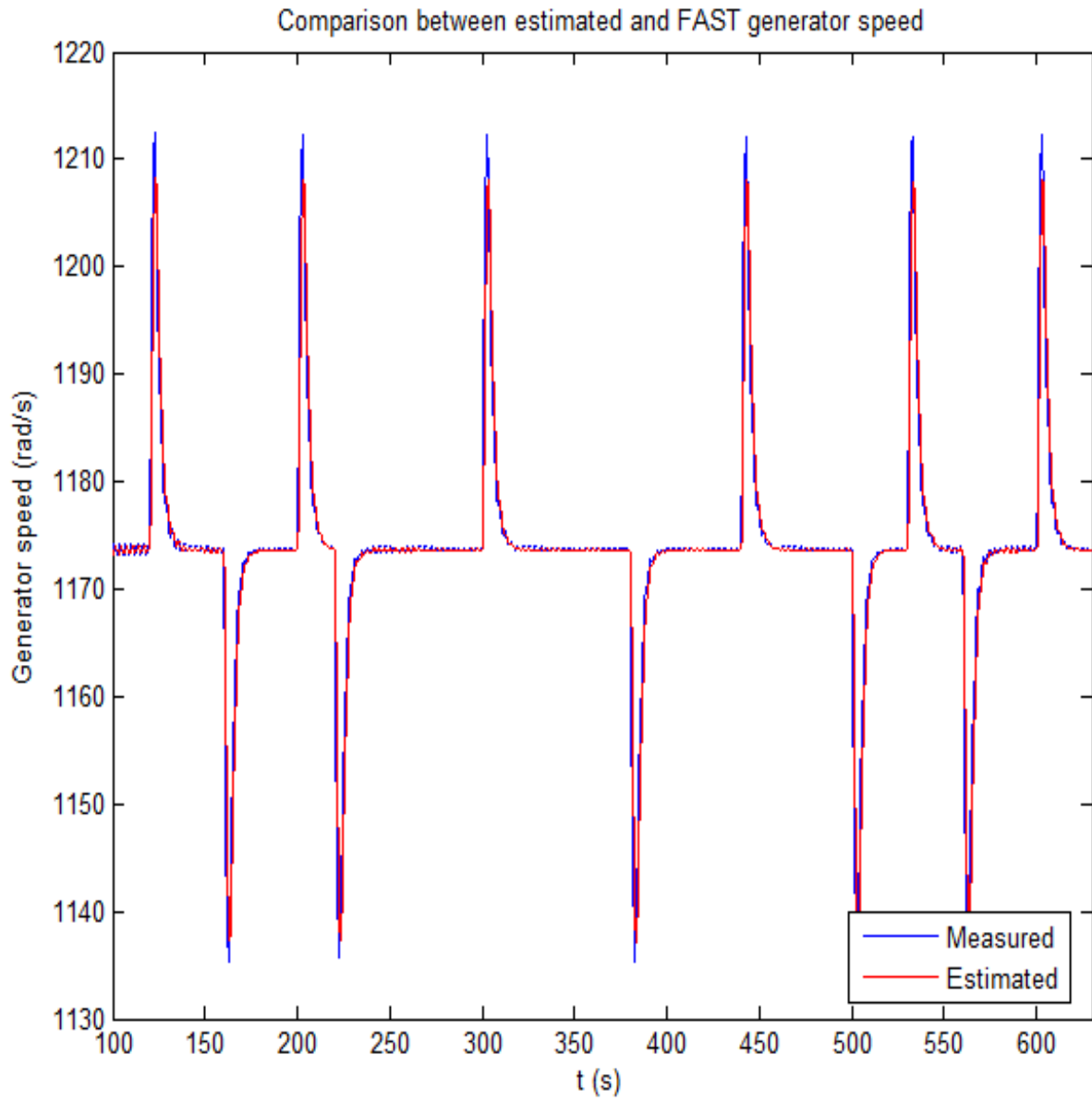Figure 4.14: Generator Torque and Generator Speed variables comparison

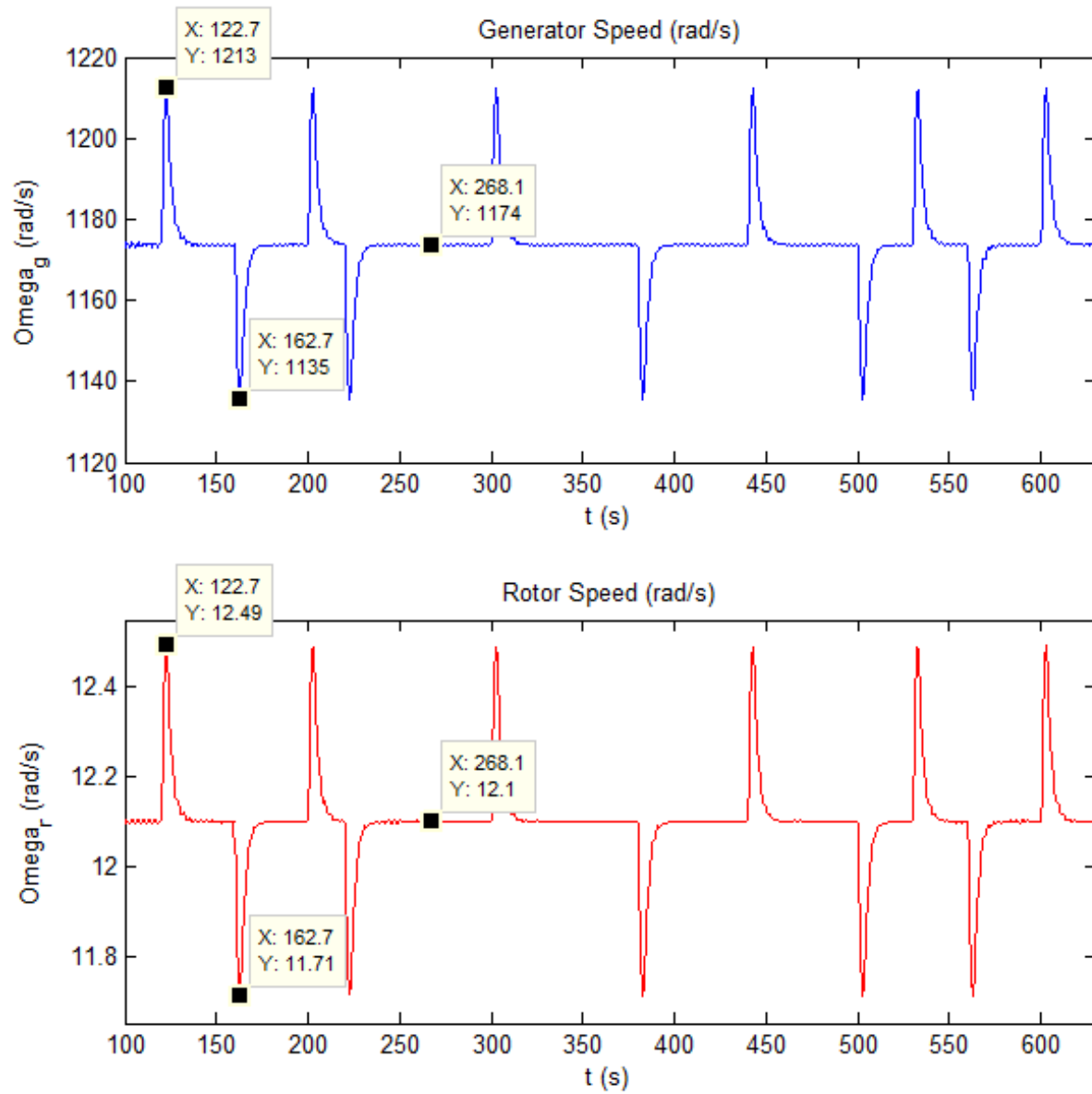Figure 4.15: Comparison between estimated and measured generator speed

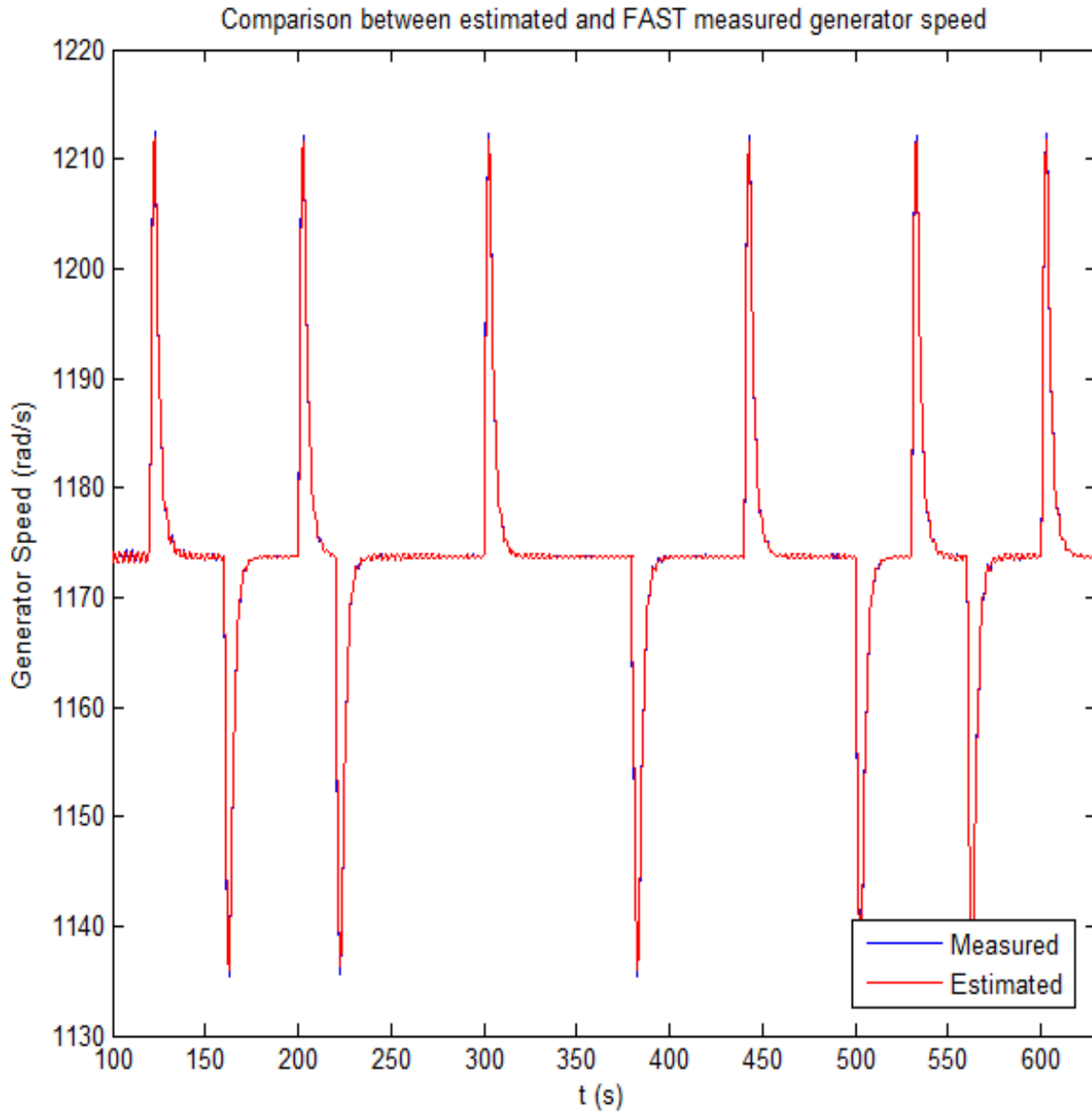Figure 4.16: Rotor Speed and Generator Speed variables comparison

Figure 4.17: Comparison between estimated and measured generator speed

4.7.1 Modified Drive Train State Space System

In the model system equations used for describing the drive train dynamics in the past section the variables involved are the generator and rotor speed, the generator and aerodynamic torque and the torsion angle. From all of these variables the only sensors available are for the rotor speed, the generator speed. In the case of the aerodynamic torque, equation (2.100), is function of the wind speed, the rotor speed and the C_p coefficient which is calculated by FAST during the simulation. The rest of the parameters of the aerodynamic torque such as the air density and the rotor swept area are also known. To obtain a model that can be used for the purpose of fault diagnosis, we neglect the dynamics of the torsion angle in the identification process because we do not have it available as a sensor.

To obtain a drive train subsystem model with variables that can be known and estimated, it is proposed a new state of space model which probed to have very good fit in the subsequent tests.

The new set of states is the following:

$$x = \begin{bmatrix} \omega_r & \omega_g \end{bmatrix}^T \tag{4.29}$$

$$y = \begin{bmatrix} \omega_r & \omega_g \end{bmatrix}^T \tag{4.30}$$

$$u = \begin{bmatrix} T_a & T_g \end{bmatrix}^T \tag{4.31}$$

$$\dot{x} = \begin{bmatrix} \frac{-1}{J_r}(B_{dt} + B_r) & \frac{B_{dt}}{J_r N_g} \\ \frac{B_{dt}}{J_g N_g} & -\frac{1}{J_g}\left(\frac{B_{dt}}{N_g^2} + \frac{B_g}{J_g}\right) \end{bmatrix} x + \begin{bmatrix} \frac{1}{J_r} \\ -\frac{1}{J_g} \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \tag{4.32}$$

Applying Laplace transform to the state space equations of system 4.32 the following equations are obtained:

$$s\omega_r(s) = \frac{-1}{J_r}(B_{dt} + B_r)\,\omega_r(s) + \frac{B_{dt}}{J_r N_g}\omega_g(s) + \frac{1}{J_r}T_a(s) \tag{4.33}$$

$$s\omega_g(s) = \frac{B_{dt}}{J_g N_g}\omega_r(s) - \frac{1}{J_g}\left(\frac{B_{dt}}{N_g^2} + \frac{B_g}{J_g}\right)\omega_g(s) - \frac{1}{J_g}T_g(s) \tag{4.34}$$

From equations (4.33) and (4.34) we obtain the reference of the structure models to be used in the identification process.

Equation (4.34) can be rewritten as:

$$\omega_g(s) = \frac{1}{s}\left(\frac{B_{dt}}{J_g N_g}\omega_r(s) - \frac{1}{J_g}\left(\frac{B_{dt}}{N_g^2} + \frac{B_g}{J_g}\right)\omega_g(s) - \frac{1}{J_g}T_g(s)\right) \tag{4.35}$$

After substituting (4.35) in (4.33) and grouping the terms the equation (4.36) is obtained

$$s^2\omega_r(s) + \frac{1}{J_r}(B_{dt} + B_r)\,s\omega_r(s) - \frac{B_{dt}^2}{J_r J_g N_g^2}\omega_r(s) = -\frac{B_{dt}}{J_g J_r N_g}\left(\frac{B_{dt}}{N_g^2} + \frac{B_g}{J_g}\right)\omega_g(s) - \frac{B_{dt}}{J_r J_g N_g}T_g(s) + \frac{1}{J_r}sT_a(s) \tag{4.36}$$

Using the transformation $s = \frac{z-1}{T_s}$ and after multiplying by z^{-2}, T_s^2 and grouping terms we obtain equation (4.37)

$$a\omega_r(z) + bz^{-1}\omega_r(z) + cz^{-2}\omega_r(z) = dz^{-2}\omega_g(z) + ez^{-2}T_g(z) + f\left(z^{-1} - z^{-2}\right)T_a(z) \tag{4.37}$$

where:

$$a = 1$$
$$b = -2 + \frac{T_s^2}{J_r}\left(B_{dt} + B_r\right)$$
$$c = 1 - \frac{T_s^2}{J_r}\left(B_{dt} + B_r\right) - \frac{T_s^2 B_{dt}^2}{J_r J_g N_g^2}$$
$$d = -\frac{T_s^2 B_{dt}}{J_g J_r N_g}\left(\frac{B_{dt}}{N_g^2} + \frac{B_g}{J_g}\right)$$
$$e = -\frac{T_s^2 B_{dt}}{J_r J_g N_g}$$
$$f = \frac{T_s}{J_r}$$

(4.38)

To obtain an identified model of drive train subsystem we apply the perturbation on the generator torque and we consider the following system with two inputs, the generator torque and the aerodynamic torque with the output the generator speed.

During the identification process several tests were made, taking one or two inputs and one output of the studied structural relation. The model structure that gave a better fit and model output is detailed below.

Inputs: $(T_a(t), T_g(t))$. **Output** $\omega_r(t)$

In this test, the best fit was obtained using a discrete time ARX model which is showed below.

$$A(z)y(t) = B(z)u(t) + e(t)$$
$$A(z) = 1 - 1.997z^{-1} + 0.9973z^{-2}$$
$$B1(z) = 2.241e - 07z^{-1} - 2.235e - 07z^{-2}$$
$$B2(z) = -0.002272z^{-1} + 0.002266z^{-2}$$

(4.39)

The other one of the considered equations which describes the drive train is equation (4.34). In this equation the variables involved are the generator speed, the rotor speed, the aerodynamic torque and the generator torque.

Equation (4.33) can be rewritten as:

$$\omega_r(s) = \frac{1}{s}\left(\frac{-1}{J_r}\left(B_{dt} + B_r\right)\omega_r(s) + \frac{B_{dt}}{J_r N_g}\omega_g(s) + \frac{1}{J_r}T_a(s)\right)$$

(4.40)

After substituting (4.40) in (4.34) and grouping the terms the equation (4.41) is obtained.

$$s^2\omega_g(s) + \frac{1}{J_g}\left(\frac{B_{dt}}{N_g^2} + \frac{B_g}{J_g}\right)s\omega_g(s) - \frac{B_{dt}^2}{J_g J_r N_g^2}\omega_g(s) = -\frac{B_{dt}}{J_g J_r N_g}\left(B_{dt} + B_r\right)\omega_r(s) + \frac{B_{dt}}{J_g J_r N_g}T_a(s) - \frac{1}{J_g}sT_g(s)$$

(4.41)

Using the transformation $s = \frac{z-1}{T_s}$ to discrete time where T_s is the sampling time and after multiply by z^{-2}, T_s^2 and grouping terms we obtain equation (4.42).

$$a\omega_g(z) + bz^{-1}\omega_g(z) + cz^{-2}\omega_g(z) = dz^{-2}\omega_r(z) + ez^{-2}T_a(z) + f\left(z^{-1} - z^{-2}\right)T_g(z)$$

(4.42)

where:

$$a = 1$$
$$b = -2 + \frac{T_s}{J_g}\left(\frac{B_{dt}}{N_g^2} + \frac{B_g}{J_g}\right)$$
$$c = 1 - \frac{T_s}{J_g}\left(\frac{B_{dt}}{N_g^2} + \frac{B_g}{J_g}\right) - \frac{T_s^2 B_{dt}^2}{J_g J_r N_g^2}$$
$$d = -\frac{B_{dt} T_s^2}{J_g J_r N_g}\left(B_{dt} + B_r\right)$$
$$e = \frac{B_{dt} T_s^2}{J_g J_r N_g}$$
$$f = -\frac{T_s}{J_g}$$

(4.43)

To obtain an identified model of this equation we apply the perturbation on the pitch angle which influences the rotor speed and therefore the aerodynamic torque.

During the identification process several tests are made, taking one or two inputs and one output of those implied in the equation (4.34). The model structure that gave a better fit and model output for this equation is detailed below.

Inputs: $(T_g(t), T_a(t))$. **Output** $\omega_g(t)$

In this test, the best fit was obtained using a discrete time ARX model which polynomials are showed below.

$$A(z) = 1 - 1.999z^{-1} + 0.9986z^{-2}$$
$$B1(z) = 2.216e - 05z^{-1} - 2.214e - 05z^{-2} \qquad (4.44)$$
$$B2(z) = -0.2341z^{-1} + 0.2338z^{-2}$$

Fit to estimation data: 99.04% (simulation focus). FPE: 2.707e-05, MSE: 0.007675

In figures 4.18 and 4.19 are shown the comparison between the estimated models 4.37 and 4.42 measured variables rotor speed and generator speed. It is observed a very good fit and approximation between the measured variables and the estimated models.
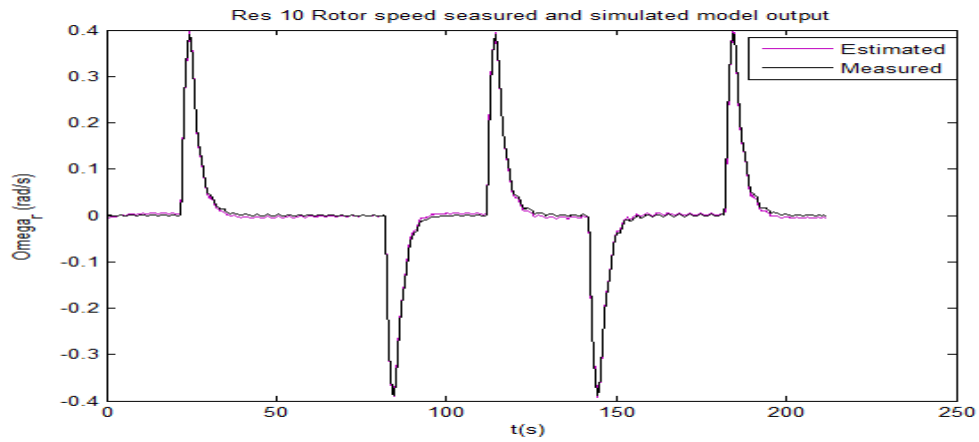

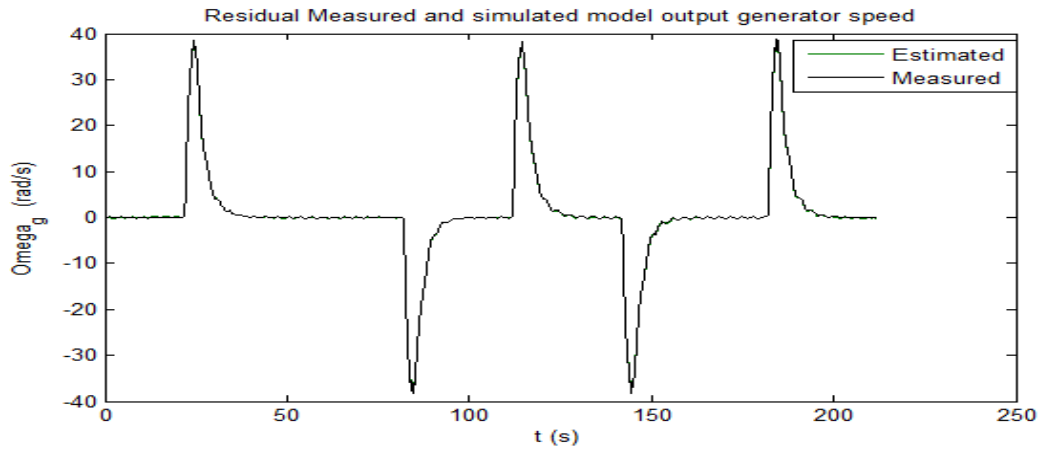Figure 4.18: Residual 10. Measured and simulated model output of generator speed

Figure 4.19: Residual 11. Measured and simulated model output of generator speed

4.8   Summary of Estimated Residuals Models

The summary of the residuals obtained from the structural analysis and those calibrated and adjusted in this chapter are shown below. These residuals are the ones that will be used for fault detection and isolation purposes as illustrated in the fault signature matrix (3.5).

Residual 1

This first analytical redundant relation is obtained directly because the power generated $P_g(t)$, the rotor speed $\omega_r(t)$ and the generator torque are all measured variables.

$$P_{\mathrm{gm}}(t) = \eta_{\mathrm{g}}\omega_{\mathrm{gm}}(t)T_{\mathrm{gm}}(t) \tag{4.45}$$

Residual 2

The ARRSs for the blade root moments are those which represented with more accuracy the dynamics obtained in FAST simulator.

$$\bar{M}_{B,1(t)} = a_2\,\bar{v}(t)^2 + a_1\,\bar{v}(t) + a_0 \tag{4.46}$$

As explained before for winds bigger than 12 m/s the following ARR is used.

$$\bar{M}_{B,1(t)} = a_2\,\bar{\beta}_1(t)^2 + a_1\,\bar{\beta}_1(t) + a_0 \tag{4.47}$$

Residual 3

$$\bar{M}_{B,2(t)} = a_2\,\bar{v}(t)^2 + a_1\,\bar{v}(t) + a_0 \tag{4.48}$$

$$\bar{M}_{B,2(t)} = a_2\,\bar{\beta}_2(t)^2 + a_1\,\bar{\beta}_2(t) + a_0 \tag{4.49}$$

Residual 4

$$\bar{M}_{B,3(t)} = a_2\,\bar{v}(t)^2 + a_1\,\bar{v}(t) + a_0 \tag{4.50}$$

$$\bar{M}_{B,3(t)} = a_2\,\bar{\beta}_3(t)^2 + a_1\,\bar{\beta}_3(t) + a_0 \tag{4.51}$$

Residual 5

$$\begin{aligned} T_g(t) &= T_{gm}(t) \\ \frac{dT_g(t)}{dt} &= \frac{dT_{gm}(t)}{dt} \end{aligned} \tag{4.52}$$

$$\tau_{\mathrm{g}}\frac{dT_{gm}(t)}{dt} + T_{gm}(t) = T_{\mathrm{g,ref}}(t) \tag{4.53}$$

Residual 6

$$\begin{aligned} \beta_{1m}(t) &= \beta_1(t) \\ \frac{d\beta_{1m}(t)}{dt} &= \frac{d\beta_1(t)}{dt} \\ \frac{d^2\beta_{1m}(t)}{dt^2} &= \frac{d^2\beta_1(t)}{dt^2} \end{aligned} \tag{4.54}$$

$$\frac{d^2\beta_{1m}(t)}{dt^2} + 2\zeta\omega_n\frac{d\beta_{1m}(t)}{dt} + \omega_n^2\beta_{1m}(t) = \omega_n^2\beta_{ref}(t) \tag{4.55}$$

Residual 7

$$\begin{aligned} \beta_{2m}(t) &= \beta_2(t) \\ \frac{d\beta_{2m}(t)}{dt} &= \frac{d\beta_2(t)}{dt} \\ \frac{d^2\beta_{2m}(t)}{dt^2} &= \frac{d^2\beta_2(t)}{dt^2} \end{aligned} \tag{4.56}$$

$$\frac{d^2\beta_{2m}(t)}{dt^2} + 2\zeta\omega_n\frac{d\beta_{2m}(t)}{dt} + \omega_n^2\beta_{2m}(t) = \omega_n^2\beta_{ref}(t) \tag{4.57}$$

Residual 8

$$\begin{aligned}
\beta_{3m}(t) &= \beta_3(t) \\
\frac{d\beta_{3m}(t)}{dt} &= \frac{d\beta_3(t)}{dt} \\
\frac{d^2\beta_{3m}(t)}{dt^2} &= \frac{d^2\beta_3(t)}{dt^2}
\end{aligned} \tag{4.58}$$

$$\frac{d^2\beta_{3m}(t)}{dt^2} + 2\zeta\omega_n \frac{d\beta_{3m}(t)}{dt} + \omega_n^2\beta_{3m}(t) = \omega_n^2\beta_{ref}(t) \tag{4.59}$$

From the new state-space model described in equation (4.32) the following ARRs are obtained.

Residual 9

$$B_g\omega_g(t) = -T_g(t) + K \tag{4.60}$$

$$f\left(\omega_{gm}(t), T_{gm}(t)\right) = 0 \tag{4.61}$$

Residual 10

$$f_1\left(\dot{\omega}_{rm}(t), \omega_{rm}(t), \omega_{gm}(t), T_a*(t), T_{gm}(t)\right) = 0 \tag{4.62}$$

$$T_{a*}(t) = f\left(\omega_{rm}(t), v_m(t), \beta_m(t)\right) \tag{4.63}$$

Residual 11

$$f_2\left(\dot{\omega}_{gm}(t), \omega_{rm}(t), \omega_{gm}(t), T_{a*}(t), T_{gm}(t)\right) = 0 \tag{4.64}$$

Residual 12

$$\omega_{rm}(t) - \frac{1}{N_g}\omega_{gm}(t) = 0 \tag{4.65}$$

Chapter 5

Fault Diagnosis System Implementation

In this chapter the implementation of the fault detection is described, and the results of it's application for each fault scenario are evaluated. The fault isolation based on the theoretical signature matrix is also commented later in this chapter.

5.1 Model Based Detection Methods

The model based detection methods are based in assessing the consistency between observations of a system and the outputs of a model. The behavior of the real process and the behavior of a model are compared, if the output of the system and the models output are different then a fault is detected. This explanation is showed in Figure 5.1.

Figure 5.1: Model Based Fault Detection Scheme
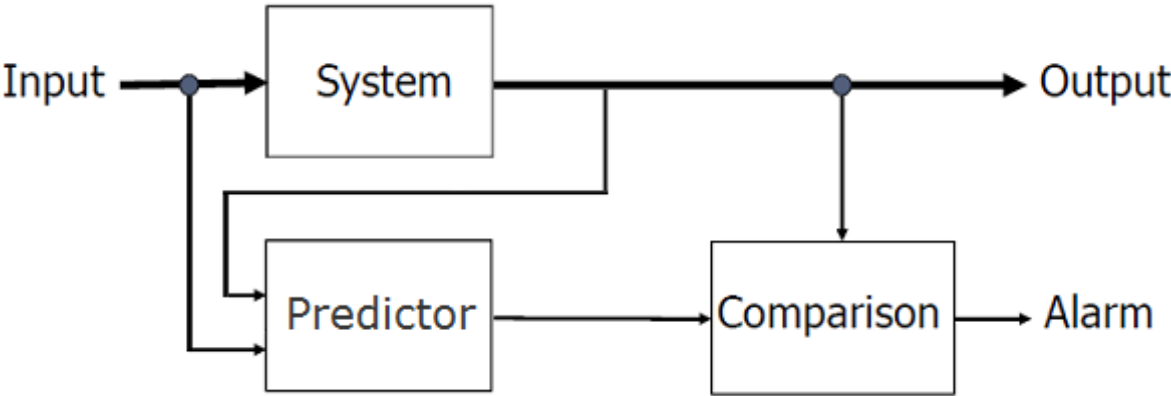
A residual is a variable used for evaluating consistency. A residual is defined as:

$$r(t) = y(t) - \hat{y}(t) \tag{5.1}$$

where: - $y(t)$ is the real process output.
- $\hat{y}(t)$ is the predictor output.

Behaviors between real and predicted outputs are always different due to the uncertainty. Uncertainty in industrial processes makes difficult the fault detection task.

Ideally residuals are 0 when there are no faults and not 0 when there are faults. But in many cases there are modeling errors, non-modeled dynamics, disturbances and noise which cause that residuals are not 0 even when there are no faults.

Model-based fault detection tests are based on the evaluation of a set of fault indication signals (residuals), obtained through analytical redundancy relations ARRs see chapter 2.

A fault detection task decides if an ARR is violated at a given instant or not generating an s_i according to:

$$s_i(k) = \begin{cases} 0, & if \quad \tau_{lower} < r_i(k) < \tau_{upper} & (no\,fault) \\ 1, & if \quad r_i(k) \leq \tau_{lower} \; or \; r_i(k) \geq \tau_{upper} & (fault) \end{cases} \tag{5.2}$$

where τ_{lower} and τ_{upper} are the limits of the threshold associated to the ARR r_i.

5.2   Detection Thresholds

In this section, it is explained how the detection thresholds were determined and the criteria used. During the first tests made, it was observed that with the activation of many of the considered faults in this thesis, changes in the complete set of residuals were observed. From the physical point of view this is logical because in the case of a wind turbine system where all the components are interconnected a change in the behavior of one component affects the rest.

This phenomena was observed in the residuals of several fault scenarios. In practice, this translates into all the residuals flags activation if the thresholds are set very restrictive and therefore obtaining a high number of false detections. Even though the existence of this phenomena it was observed that residuals sensitivity behaved according to the theoretical signature matrix. This means that the value of a residual in the presence of its theoretical respective fault was considerable, and in the case of the rest of the fault scenarios the changes in the residuals were small.

To address the problem of all the residuals activation, data from all the residuals simulating each one of the fault scenarios was taken, and for each one of the residuals were stored the maximum and the minimum values for the case of a fault that should not cause the residual flag triggering according to the theoretical signature matrix. This translates in practice that the residual activation works as the expected in theory and makes possible the fault detection and isolation.

In 5.2 are showed the maximum and minimum values for each fault scenario. The numbers in red which are the values of the residuals in the case of its theoretical fault occurrence are very high in comparison to the values in the presence of the rest faults.

The values of the thresholds for each one of the residuals, (see table 5.1), were determined based on the maximum and the minimum values in case of the rest of the fault scenarios that should not activate the theoretical residual, see (5.2). Only the faults which affect the subsystems modeled in this thesis were considered in the implementation of the fault detection system, faults f_2, f_6 and f_10 were not considered because of the explained in 3.5.2. Once the thresholds are defined, it can be deducted that the signature matrix obtained from the fault scenarios simulation will be different from the theoretical one as showed in table (5.2). This table summarizes the residuals activation and forms the signature matrix obtained from the fault detection system tested with the considered fault scenarios. It shows that fault scenario 1 is not detected due to the thresholds values chosen in section 5.2. Fault scenarios 3, 4, 5, 7 and 8 are detected according to the obtained in the theoretical signature matrix 3.5 and in fault scenario 9 all the theoretical residuals are activated except residual 11 which values were not bigger enough to violate its assigned threshold. The fault detection system can detect most of the fault scenarios proposed according to the theoretical signature matrix, there is only one fault that is not detected. It was needed to do a tradeoff in the thresholds assignment between missing some faults and having false alarms, the chosen criteria was to avoid all the false alarms.

Table 5.1: Detection thresholds determined for the set of residuals

	Thresholds	
	Upper	**Lower**
r1	42	-58
r2	52	-98
r3	55	-87
r4	54	-98
r5	$1.65*10^-6$	$-1.7*10^-6$
r6	-0.003	-0.006
r7	-0.003	-0.006
r8	-0.003	-0.006
r9	19	-26
r10	0.0056	0.0015
r11	-0.43	-0.49
r12	5	-6

Figure 5.2: Table with the maximum and minimum values for each residual in every fault scenario

		f1			f3	f4			f5	f7			f8			f9
		fMB1	fMB2	fMB3	f3	fβ1	fβ2	fβ3	f5	f7-PA1	f7-PA2	f7-PA3	f8-PA1	f8-PA2	f8-PA3	f9
r1	max	1,601166	1,601166	1,601166	2634,244	1,601166	1,601166	1,601166	4957,161	2,267257	2,428675	2,503351	38,12012	35,0994	41,69372	1150,131
	min	-1,27608	-1,27608	-1,27608	-3,74327	-1,27608	-1,27608	-1,27608	-1,27608	-1,38458	-1,5084	-1,47249	-55,4017	-57,5571	-55,83	-1150,27
r2	max	-8,4516	-5,80192	-5,80192	6,42499	-5,80192	-5,80192	-5,80192	-5,80192	109,0963	7,285127	7,724078	1385,411	51,06099	40,60048	-2,90899
	min	-22,4304	-14,9987	-14,9987	-1,69274	-6209,33	-14,9987	-14,9987	-14,9987	-72,7714	-28,0172	-29,9146	-1375,22	-97,1332	-84,591	-19,1807
r3	max	-5,9698	-8,65656	-5,9698	8,12456	-5,9698	-5,9698	-5,9698	-5,9698	6,924738	109,991	9,671269	54,51271	1386,785	39,95222	-3,26447
	min	-15,7385	-22,5517	-15,7385	-1,31953	-15,7385	-6208,51	-15,7385	-15,7385	-31,1351	-71,8673	-30,3733	-86,9722	-1373,56	-83,0645	-19,3998
r4	max	-5,43239	-5,43239	-8,99364	10,23868	-5,43239	-5,43239	-5,43239	-5,43239	6,397961	7,823183	109,8762	53,69123	49,44093	1386,042	-2,74757
	min	-16,7081	-16,7081	-22,8595	0,348551	-16,7081	-16,7081	-6213,05	-16,7081	-29,3432	-26,8823	-69,0408	-89,0715	-97,9092	-1376,55	-19,2399
r5	max	3,81E-08	3,81E-08	3,81E-08	1,63E-06	3,81E-08	3,81E-08	3,81E-08	3,81E-08	6,14E-08	6,44E-08	7,00E-08	6,67E-07	7,66E-07	7,07E-07	1
	min	-3,89E-08	-3,89E-08	-3,89E-08	-1,68E-06	-3,89E-08	-3,89E-08	-3,89E-08	-3,89E-08	-4,53E-08	-3,89E-08	-3,95E-08	-5,18E-07	-5,10E-07	-4,42E-07	-0,536
r6	max	-0,0052	-0,0052	-0,0052	-0,00493	20,68234	-0,0052	-0,0052	-0,0052	0,032436	-0,00518	-0,00518	0,747741	-0,00508	-0,0051	-0,00519
	min	-0,00521	-0,00521	-0,00521	-0,00525	-21,0607	-0,00527	-0,00521	-0,00521	-0,01532	-0,00523	-0,00523	-0,20757	-0,00529	-0,00527	-0,00521
r7	max	-0,0052	-0,0052	-0,0052	-0,00493	-5,20E-03	20,68234	-5,20E-03	-0,0052	-0,00518	0,032439	-0,00518	-0,0051	0,74768	-0,0051	-0,00519
	min	-0,00521	-0,00521	-0,00521	-0,00525	-0,00521	-21,0607	-0,00521	-0,00521	-0,00523	-0,01532	-0,00523	-0,00529	-0,20757	-0,00527	-0,00521
r8	max	-0,0052	-0,0052	-0,0052	-0,00493	-5,20E-03	-0,0052	2,07E+01	-0,0052	-0,00518	-0,00518	0,032433	-0,0051	-0,00508	0,747525	-0,00519
	min	-0,00521	-0,00521	-0,00521	-0,00525	-0,00521	-0,00527	-21,0607	-0,00521	-0,00523	-0,00523	-0,01532	-0,00529	-0,00529	-0,20757	-0,00521
r9	max	1,332527	1,332527	1,332527	61,6075	1,332527	1,332527	1,332527	1,332527	1,54806	1,332527	1,352025	18,28208	17,99621	15,52358	27,9399
	min	-1,30779	-1,30779	-1,30779	-58,8046	-1,30779	-1,30779	-1,30779	-1,30779	-2,10315	-2,20347	-2,40469	-22,2766	-25,4578	-23,5977	-1,30779
r10	max	0,003653	0,003653	0,003653	0,00391	0,003653	0,003653	0,003653	0,003653	0,00368	0,00367	0,003697	0,005365	0,005511	0,005366	0,005857
	min	0,003605	0,003605	0,003605	0,003541	0,003605	0,003605	0,003605	0,003605	0,003562	0,003561	0,003554	0,001963	0,00168	0,001944	0,001389
r11	max	-0,46452	-0,46452	-0,46452	61,3051	-0,46452	-0,46452	-0,46452	-0,46452	-0,46329	-0,46358	-0,46348	-0,44574	-0,44351	-0,4453	-0,45202
	min	-0,46699	-0,46699	-0,46699	-62,1742	-0,46699	-0,46699	-0,46699	-0,46699	-0,46699	-0,46699	-0,46699	-0,48277	-0,48423	-0,48168	-0,47853
r12	max	0,057663	0,057663	0,057663	0,534314	0,057663	0,057663	0,057663	0,057663	0,130837	0,140951	0,185426	4,282681	4,059931	3,921793	0,856434
	min	-0,09637	-0,09637	-0,09637	-61,7925	-0,09637	-0,09637	-0,09637	-0,09637	-0,13591	-0,13644	-0,16237	-4,89731	-5,25534	-4,99863	-0,85874

Table 5.2: Signature matrix obtained from simulating the fault scenarios

	f_1			f_3	f_4			f_5	$f_{(7-8)}$			f_9
	$f_{MB,1,m}$	$f_{MB,2,m}$	$f_{MB,3,m}$		$f_{4-\beta1}$	$f_{4-\beta2}$	$f_{4-\beta3}$		$f_{(7-8)-PA1}$	$f_{(7-8)-PA2}$	$f_{(7-8)-PA3}$	
r1				x				x				x
r2					x							
r3						x			x			
r4							x				x	x
r5												x
r6					x				x			
r7						x				x		
r8							x				x	x
r9				x								x
r10				x								x
r11				x								
r12												

5.3 Fault Detection System

The Simulink interface of FAST simulator where wind turbine system is implemented is showed on Figure 5.3. In this Simulink model, in the block called "Fault Detection Block" is where all the residuals are implemented. The residuals receive all the sensor signals and they activate its respective alarms if their thresholds are violated.

In Figure 5.4 is showed the fault detection system with each one of the twelve residuals that were proposed in this work. These residuals are those enumerated in the residuals summary in Section 4.8. When the residuals fall out of the threshold, the signal value is 0, otherwise is 1. This means that if a fault occurs and this causes a residual threshold violation the signal value will be zero.

The fault detection algorithm is running and checking in every sample if the residuals thresholds are violated. A flag variable is designated for each residual, in the case a residual threshold is violated its respective flag will be updated to value 1.

It is necessary to mention that because the residuals are tested in FAST simulator and every simulator has a initialization stage, the measured variables and therefore the residuals take some time to enter in steady state regime in which it is simulated the wind turbine to be operating with a constant rotor speed and in control zone 2 where the control strategy is to track a generated power reference. This is the reason why there are some time conditions for each residual in order to activate its respective residual flag as it is showed in the fault detection algorithm in Appendix A. The residuals implementation in simulink is illustrated in Appendix C.
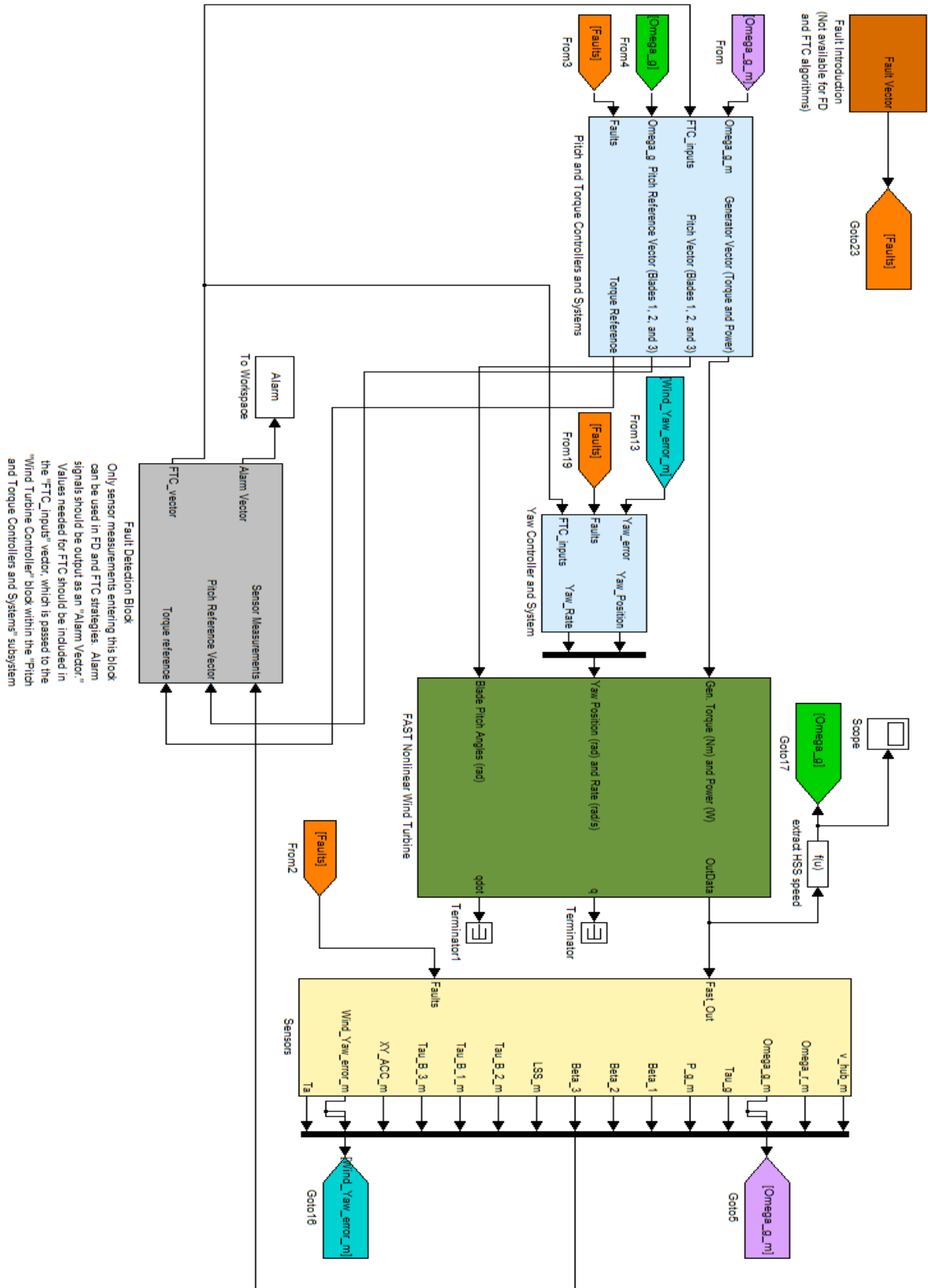
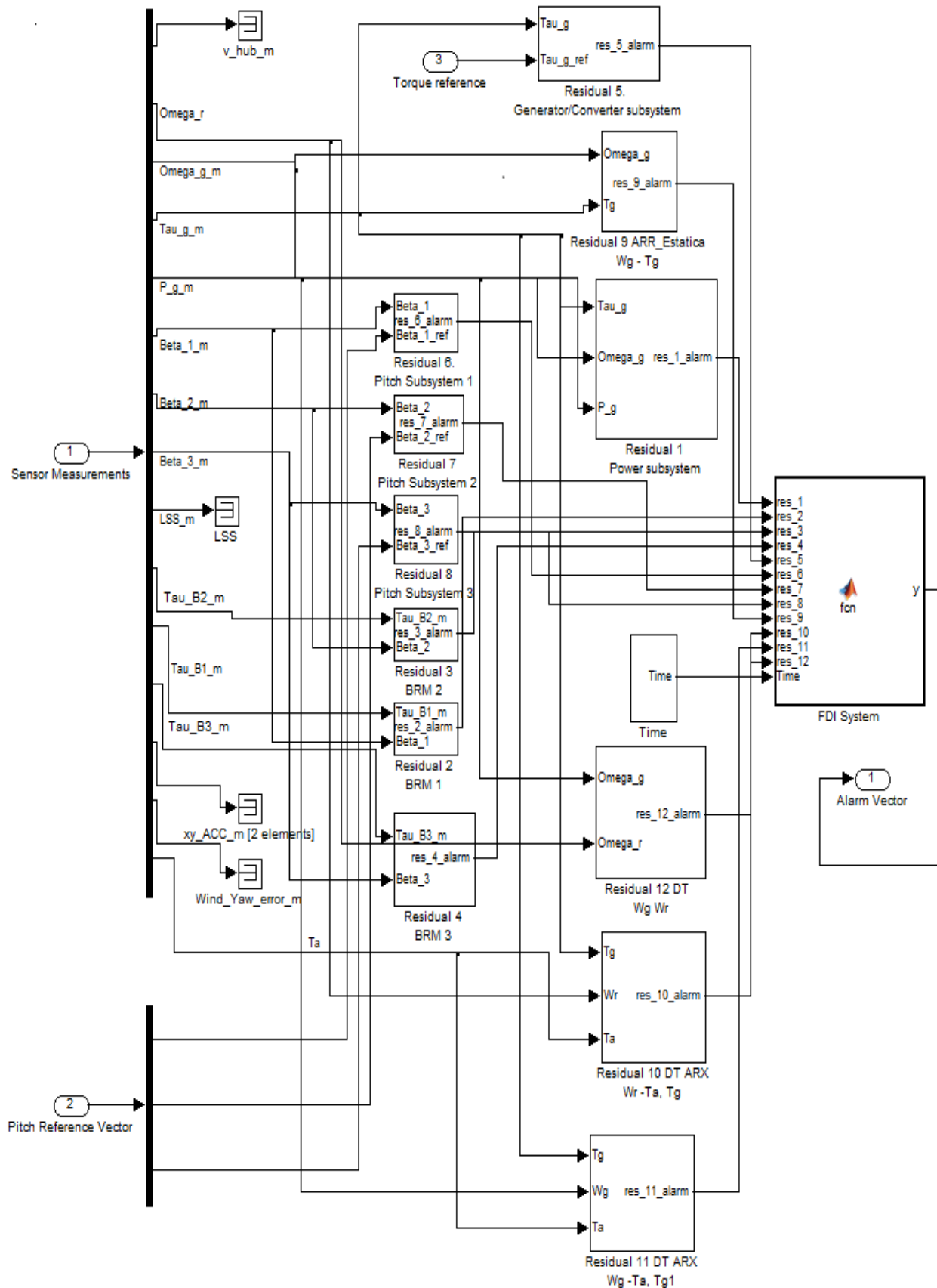Figure 5.3: Wind Turbine System in Simulink

Figure 5.4: Fault Detection System. Set of residuals implementation

5.4   Fault Scenarios Tests

In this section it is shown the residuals behavior in each fault scenario. The fault detection system performance with the presence of each fault scenario is analyzed and the results are assessed. One fault scenario at a time is presented and the behavior of the set of proposed residuals in this thesis is analyzed. The bit value for the residuals when there is no fault is 1, when the residual violates its designated threshold the value is 0 as can be seen in the figures presented for each fault scenario.

5.4.1   Fault Scenario 1

As indicated in section 3.5, the fault scenario 1 occurs when the blade root moment sensor is scaled by a factor of 0.95, this fault is present in the interval 295-320 s. Fault scenario 1 is reproduced and simulated for each one of the blades as detailed below.

Residual 2 for fault 1 on blade root moment 1 is shown in Figure 5.5. In this figure it can be seen that when fault occurs the residual is sensitive and scales the value in which the residual is converging. Residual 3 for fault 1 on blade root moment 2 and residual 4 for fault 1 on blade root moment 3 both showed the same sensitivity to their respective residuals but were not detected because of the threshold selection explained in Section 5.2.

As observed in the cases of blade root moment 1, 2 and 3, the residuals are sensitive to the fault scenario 1, even though they are not zero as expected due to modeling errors and uncertainties. However, fault scenario 1 is not detected as shown in the results of the detection in table 5.2 because the detection thresholds are bigger than the deviation caused by this fault in the blade root moment residuals. The thresholds as explained in Section 5.2, were selected according to the worst case scenario observed in presence of the other faults to avoid false detections and to prevent the phenomena of all the residuals activation explained in that section.
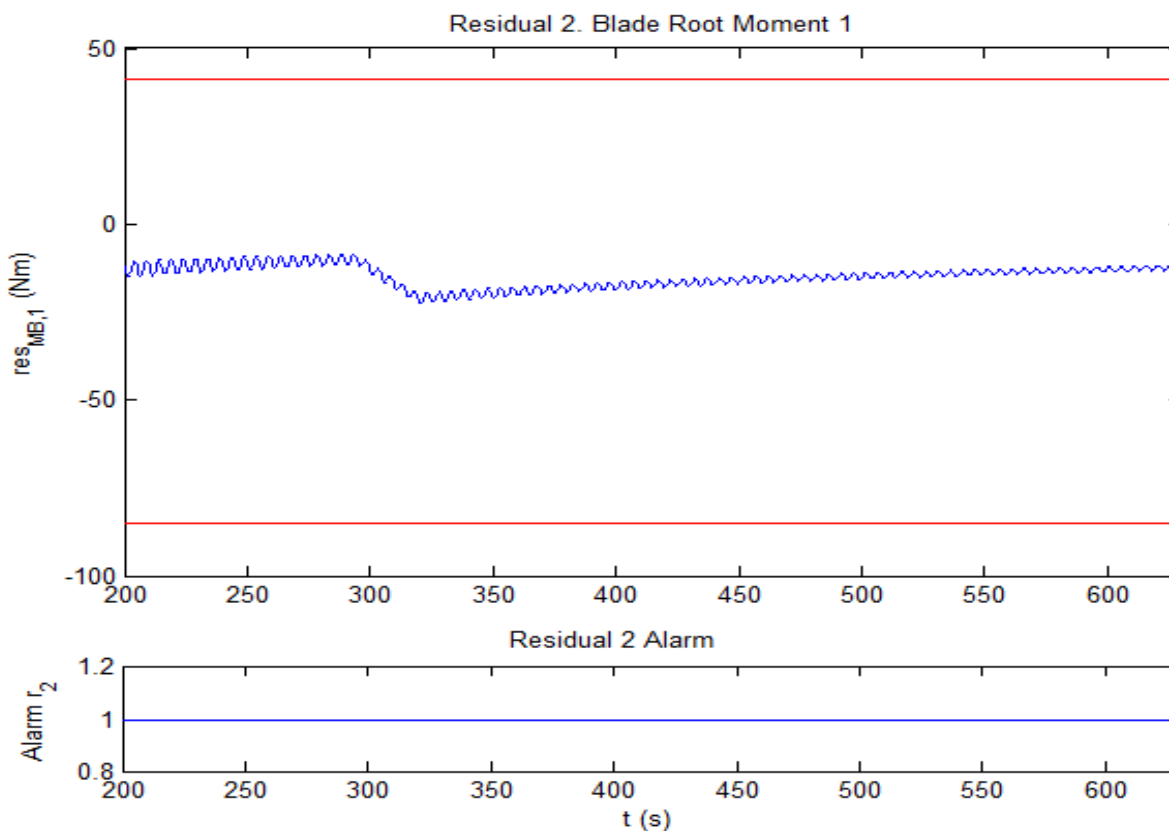

Figure 5.5: Fault 1 Blade Root Moment on Blade 1 residual

5.4.2 Fault Scenario 3

Fault scenario 3 causes the generator speed sensor to be scaled by a factor of 0.95 and it is present between 130s and 155s as explained in Section 3.5. Residuals 1, 9, 11 and 12 for fault 3 are showed in Figure 5.6. As can be seen in the figure all the theoretical residuals corresponding to this fault scenario are sensitive to this fault. The threshold set shown in Section 5.2 for this fault scenario, cause the activation of all the theoretical residuals indicated in the theoretical signature matrix (3.5), and shown in Table 5.2 of activated residuals during the fault tests. It is observed that when fault 3 occurs, all the residuals thresholds are violated and only two of them maintain themselves out of the thresholds limits during the whole fault activation.

There are two ways to maintain active the detection, one is using simulation models with the inconvenient that it has or the other one is to activate the alarm when the residual is detected as abnormal and maintaining it, this second case was the one implemented as can be seen in the fault detection algorithm shown in Appendix A.
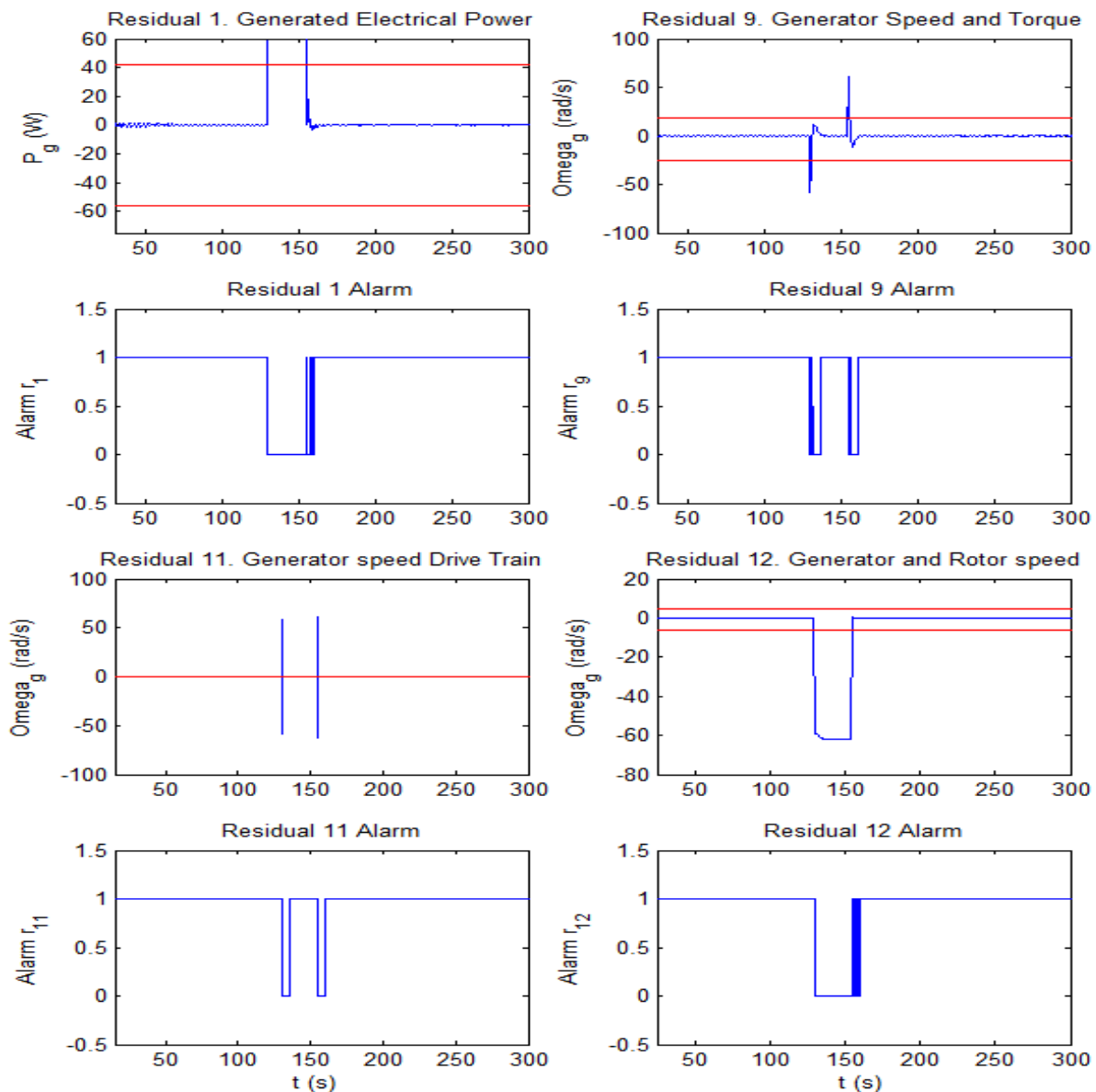
Figure 5.6: Fault 3 on Generator Speed Sensor residuals

5.4.3   Fault Scenario 4

The fault scenario 4 occurs when the blade pitch sensor is stuck during the interval of 130 s and 155s, holding a constant value of 1 deg. Fault scenario 4 is reproduced and simulated for each one of the pitch subsystems as detailed below.

Residuals 2 and 6 for fault 4 on pitch sensor 1 are shown in Figure 5.7. The figure shows that the theoretical residuals are sensitive to the fault scenario 4. Residuals 3 and 7 for fault 4 on pitch sensor 2, and residuals 4 and 8 for fault 4 on pitch sensor 3 were also sensitive to the fault scenario 4 activating its respective alarms in the same way.
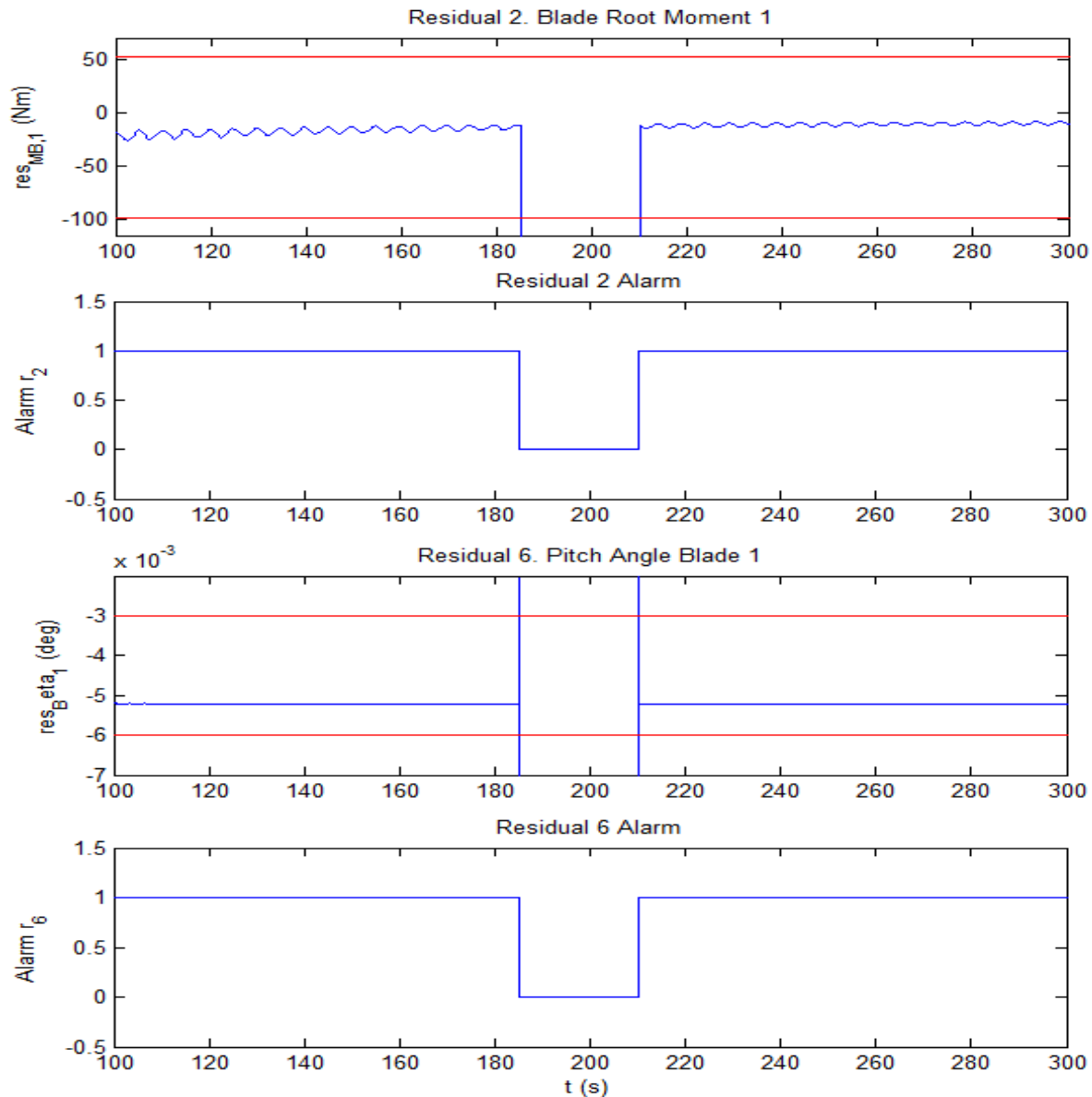
Figure 5.7: Fault 4 on Pitch Sensor 1 residuals

As observed in this section, fault scenario 4 was reproduced for the pitch subsystems 1, 2 and 3, showing in each case that their respective theoretical residuals are sensitive to this fault. The thresholds set shown in Section 5.2 for fault scenario 4, in each one of the pitch subsystems results in the activation of all the theoretical residuals indicated in Table (3.5), and shown in Table 5.2.

5.4.4 Fault Scenario 5

Fault scenario 5 as indicated in Section 3.5, corresponds to the scaling of the generator power sensor by a factor of 1.1. Fault 5 is present in the time interval from 240s to 265s.

Residual 1 for fault 5 on power sensor is shown in Figure 5.8. This figure shows that the residual is sensitive to this fault. In this case there is only one residual sensitive to this fault scenario according to the fault signature matrix (3.5) and in Table 5.2, it is shown that this residual is activated during the fault test.
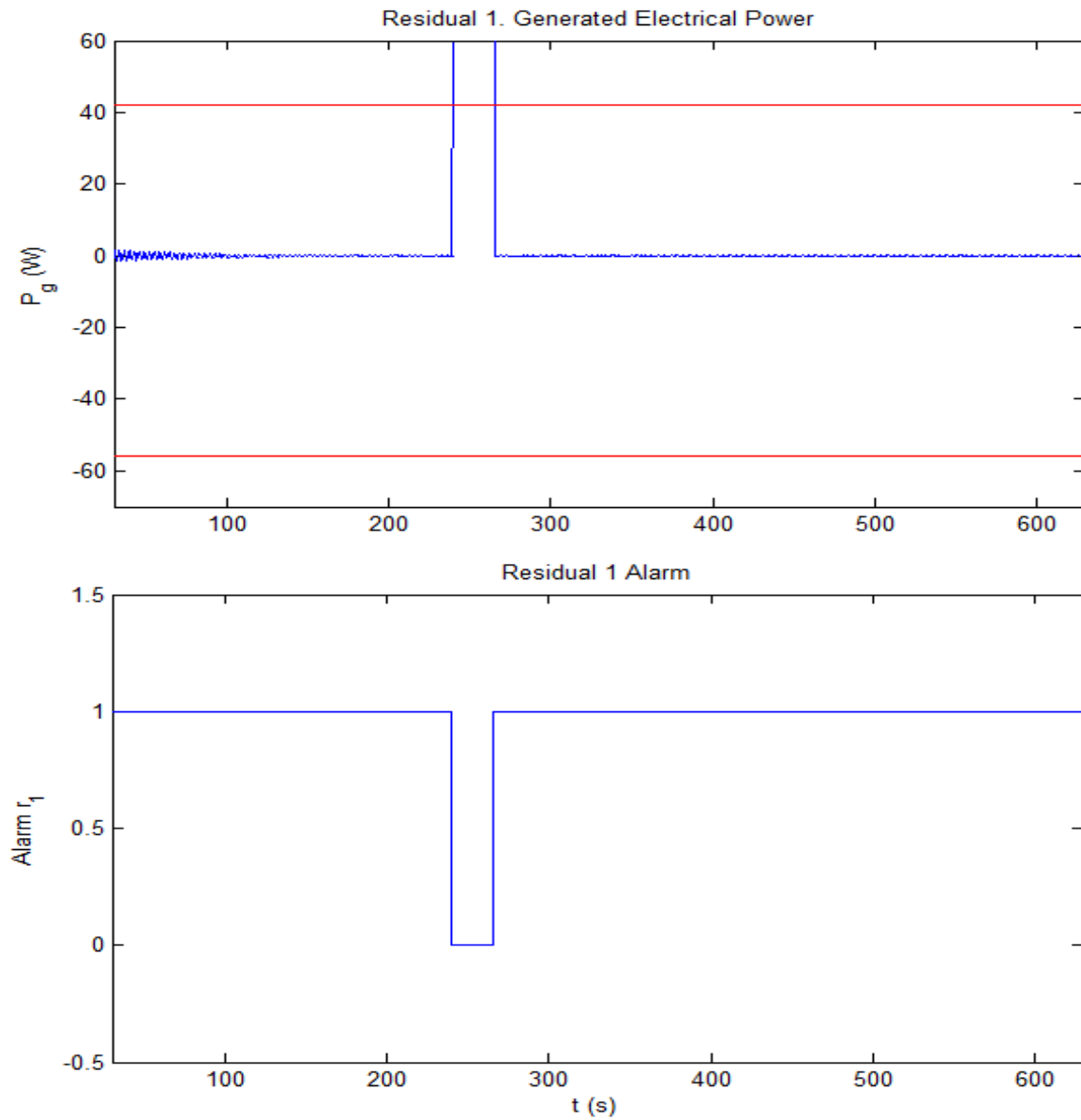
Figure 5.8: Fault 5 Power Sensor residual

5.4.5   Fault Scenario 7

The fault scenario 7 as detailed in Section 3.5 corresponds to a pitch actuator fault modeled by a change of the parameters in the pitch subsystem model. This fault is introduced linearly from 350s to 370s, and full active from 370s to 390s and linearly outfaced from 390s to 410s. Fault 7 is reproduced and simulated for each one of the pitch subsystems as detailed below.

Residuals 2 and 6 for fault 7 on pitch actuator 1 are showed in Figure 5.9, where it can be seen that the residuals are sensitive to this fault. Residuals 3 and 7 for fault scenario 7 on pitch actuator 2 and residuals 4 and 8 for fault 7 on pitch actuator 3 were also sensitive to fault scenario 7 activating the residuals in the same way.
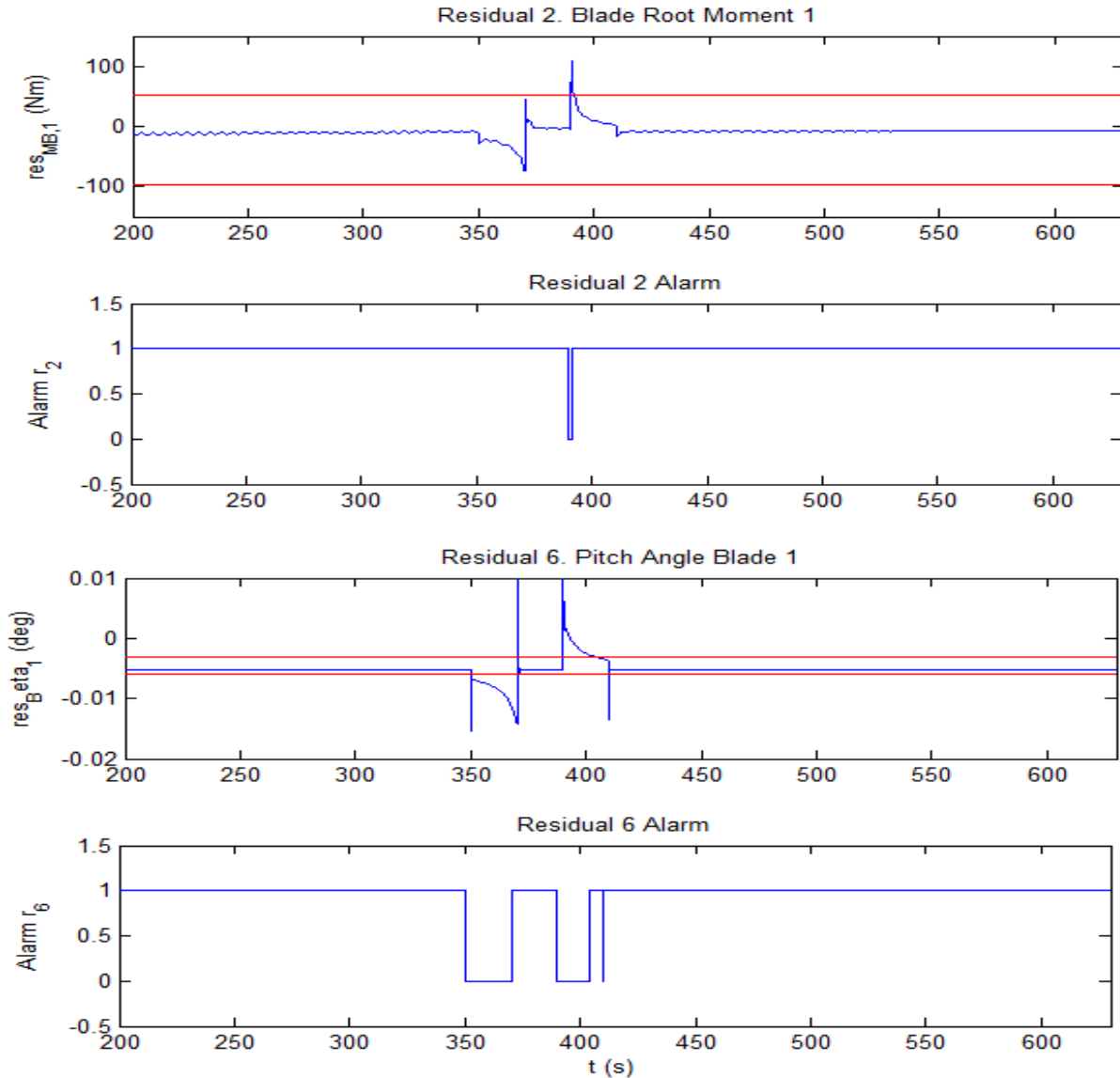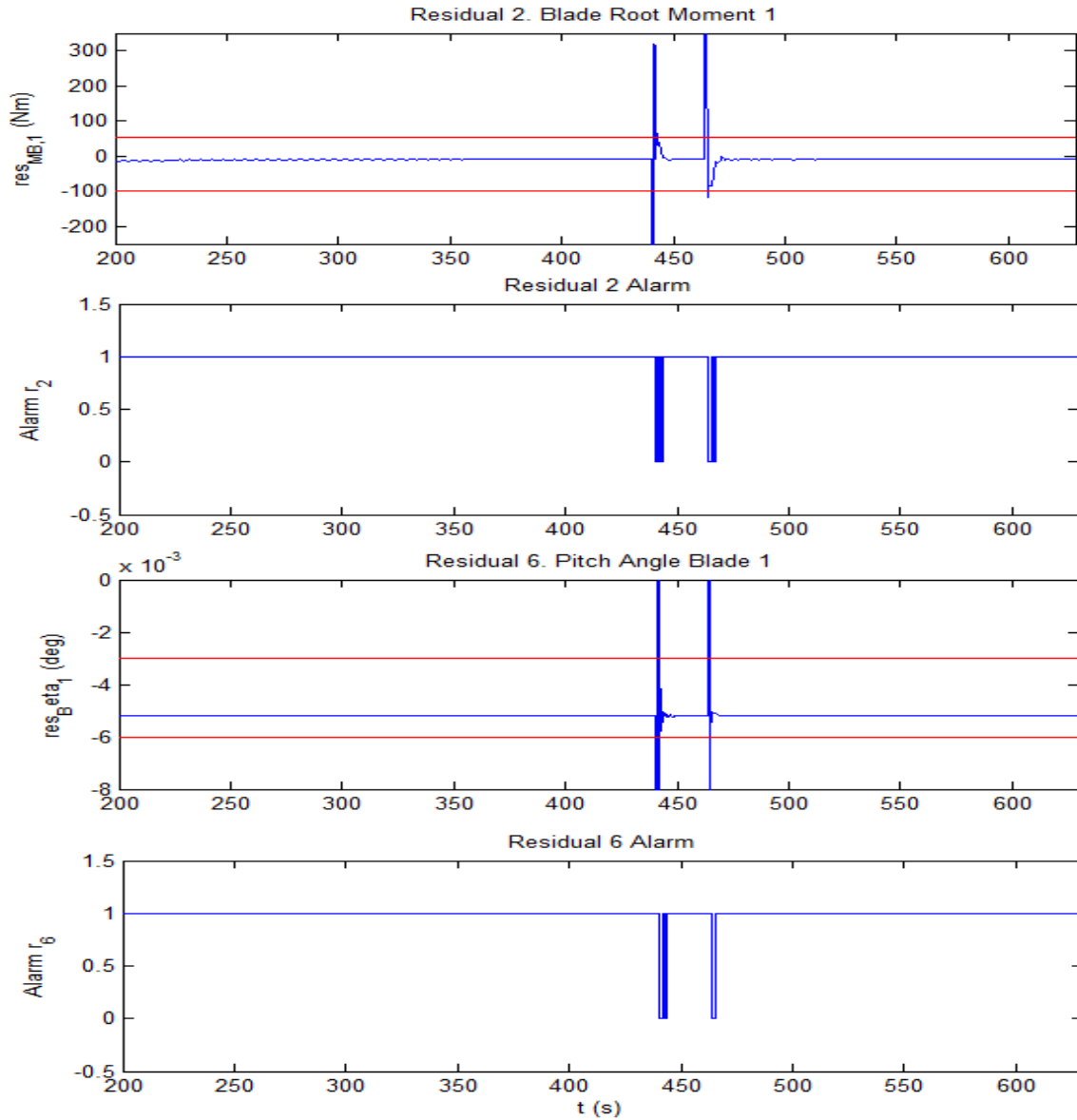

Figure 5.9: Fault 7 Pitch Actuator 1 residuals

As observed in this section, fault scenario 7 was reproduced for the pitch subsystems 1, 2 and 3, showing in each case that their respective theoretical residuals are sensitive to this fault. The thresholds set showed in Section 5.2 for fault scenario 7, in each one of the pitch subsystems results in the activation of all the theoretical residuals indicated in Table (3.5), and shown in Table 5.2.

5.4.6 Fault Scenario 8

The fault scenario 8 as detailed in Section 3.5 corresponds to a pitch actuator fault modeled by a change of the parameters in the pitch subsystem model. Fault 8 is active from 440s to 465s, and linearly introduced and outfaced within 1s both. Residuals 2 and 6 for fault scenario 8 on pitch actuator 1 are showed in Figure 5.10. Residuals 3 and 7 for fault scenario 8 on pitch actuator 2 and residuals 4 and 8 for fault scenario 8 on pitch actuator 3 were also sensitive to this fault scenario activating its respective residuals alarms in the same way.

Figure 5.10: Fault 8 Pitch Actuator 1 residuals

As observed in this section, fault scenario 8 was reproduced for the pitch subsystems 1, 2 and 3, showing in each case that their respective theoretical residuals are sensitive to this fault. The thresholds set shown in Section 5.2 for fault scenario 8, in each one of the pitch subsystems results in the activation of all the theoretical residuals indicated in Table (3.5), and shown in Table 5.2.

5.4.7   Fault Scenario 9

Fault scenario 9 as indicated in Section 3.5 corresponds to an offset on the generator torque, and occurs from 495s-520s.

    Residuals 1, 5, 9, 10 and 11 for fault 9 on pitch actuator 3 are presented in Figure 5.11, which shows that all the residuals are sensitive to this fault scenario. In this case not all the residuals flags are activated because of the thresholds limits introduced in 5.2, the activated residuals in this fault scenario were residuals 1, 5, 9 and 10 and presented in Table 5.2 of activated residuals during the fault tests. Residual flag 11 was not activated because the determined thresholds, were bigger than the change caused by fault 9 in this residual.
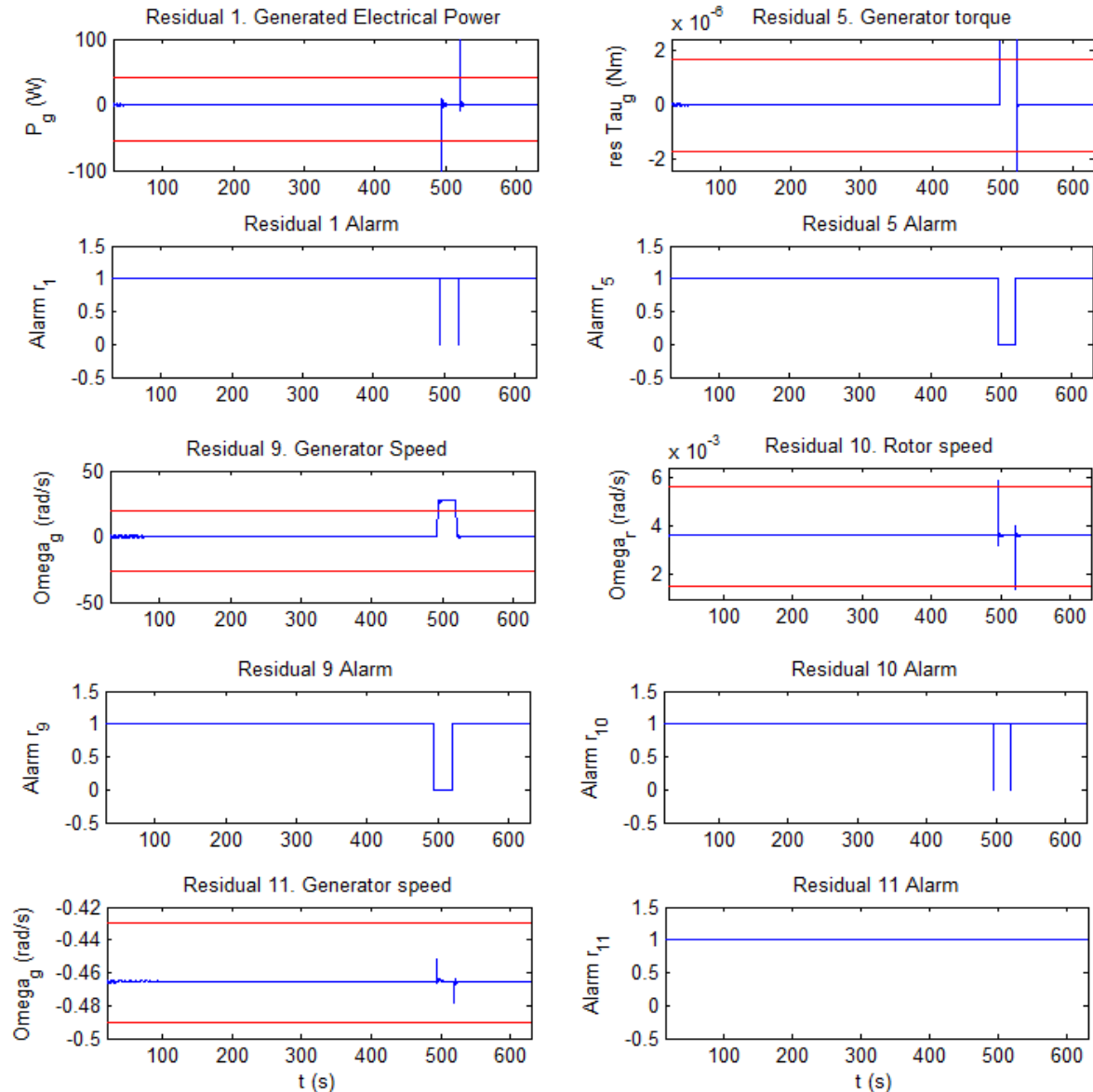
Figure 5.11: Fault 9 Generator Torque Offset residuals

5.5 Fault Isolation

After a fault isolation analysis is done, it is observed that the theoretical signature matrix 3.5 does not match with the programmed alarms indicators of the fault detection system designed, this traduces into difficulties at the time of performing a diagnosis. In practice, it is not possible to do isolations tests with a wind turbine, therefore the fault isolation is done based on the theoretical signature matrix. In this section fault isolation analysis based on column reasoning (FDI) and row reasoning (DX), introduced before in section 3.6 is done to the results obtained by the fault detection system and summarized in table 5.2. The results obtained by both techniques is analyzed and compared.

FDI

Performing column reasoning to the signature matrix obtained in fault scenarios test and shown in Section 5.2, the conclusions will be the following.

- **Fault scenario 1** was not detected at all, as a consequence this scenario could not be isolated.

- **Fault scenario 3** can be completely isolated applying column reasoning, because its fault indicators remain active. The result of the diagnosis is f_3.

- **Fault scenario 5**, can also be correctly isolated applying column reasoning because all the activated residuals correspond to those obtained in the theoretical signature matrix. The result of the diagnosis is f_5.

- **Fault scenarios 4, 7 and 8** correspond to faults that occur in the sensors and actuators of the pitch subsystems, they have the same fault signature, as a consequence they cannot be isolated as explained in Section 3.6. It is necessary to mention that it is possible to locate in which pitch subsystem the fault is located, i.e., is the pitch subsystem of blade 1, 2 or 3, but unfortunately it is not possible to determine if the fault occurred in the sensor or in the actuator. The result of the diagnosis in this case will be $f_4 \vee f_7 \vee f_8$.

- **Fault scenario 9** signature does not match exactly with its respective theoretical residual because this fault is not detected by residual 11. Therefore in a strict reasoning, this fault scenario is not isolable because do not match with none of the signatures in the theoretical signature matrix. However in the case this fault scenario occurs it is possible to guess that this is the fault that is occurring because the respective fault signature obtained in simulations does not match and is not similar to the rest of the other fault signatures.

DX

Performing row reasoning to the signature matrix obtained in fault scenarios test and shown in Section 5.2, the conclusions will be the following.

- **Residual 1 activation**. In the case of activation of r_1, the possible faults would be f_3, f_5 or f_9.

- **Residual 2 or 6 activation**. In the case of r_2 or r_6 activation, the fault would be $f_{4-\beta1} \vee f_{7-PA1} \vee f_{8-PA1}$, detecting a fault in pitch subsystem 1 but not being able to isolate if it is the sensor or the actuator the faulty component.

- **Residual 3 or 7 activation**. If r_3 or r_7 activates the fault would be $f_{4-\beta2} \vee f_{7-PA2} \vee f_{8-PA2}$, detecting a fault in pitch subsystem 2 but not being able to isolate if it is the sensor or the actuator the faulty component.

- **Residual 4 or 8 activation**. In the case of r_4 or r_8 activation, the fault would be $f_{4-\beta3} \vee f_{7-PA3} \vee f_{8-PA3}$, detecting a fault in pitch subsystem 3 but not being able to isolate if it is the sensor or the actuator the faulty component.

- **Residual 5 activation**. If r_5 activates, fault 9 would be isolated.

- **Residual 9 activation**. In the case of r_9 activation the possible faults would be f_3 or f_9.

- **Residual 10 activation**. The case of r_{10} activation would result in f_9 isolation.

- **Residual 11 or 12 activation**. If r_{11} or r_{12} activates the isolated fault would be f_3.

FDI and DX comparison

The analysis derived of FDI column reasoning leaves two fault scenarios f_3 and f_5 completely isolated. Regarding fault scenarios 4, 7 and 8, since the signatures correspond to their theoretical ones, it is possible to determine the faulty pitch subsystem but it is not possible to isolate if the faulty component is the sensor or the actuator. Fault scenario f_9 is not strictly isolated because the matching with its respective theoretical signature is not the same.

On the other hand DX row reasoning allows that two fault scenarios f_3 and f_9 are completely isolable. Similarly to FDI reasoning fault scenarios 4, 7 and 8 signatures correspond to their theoretical ones and allows to isolate the faulty pitch subsystem but does not makes possible to isolate if the sensor or the actuator is the faulty component. Fault scenario f_5 could be detected but not isolated. Finally, fault scenario 1 is not detected and therefore the activation of the residuals which theoretically consider it which are r_2, r_3 and r_4 are attributed to faults f_4, f_7 or f_8 according to the theoretical signature matrix.

FDI and DX reasoning obtain the same number of completely isolated faults, f_3 is correctly isolated in both cases with f_5 and f_9 only isolated by one of the techniques. A faulty pitch subsystem is isolated with both techniques but they present the limitation of not being able to detect if the faulty component is the sensor or the actuator.

Chapter 6

Conclusions

In this master thesis, a model-based fault detection system was designed and implemented to detect the proposed fault scenarios of the benchmark model considered in this thesis that corresponds to a wind turbine reference was a 5-MW NREL modeled and parameterized in FAST simulator. The study was based in models found in the literature for each one of wind turbine subsystems. In the fault diagnosis, a model-based approach was selected. First a structural analysis allows to derive the ARRs and then they are calibrated using data as well as the detection thresholds. Finally the implementation and testing of the fault detection system is carried out. The last stage of this work was to perform a fault isolation analysis applying the FDI and DX communities techniques to the results summarized in the signature matrix obtained from the simulation of the implemented fault diagnosis system in presence of the considered set of fault scenarios.

Modeling

A big effort was made to model the different wind turbine subsystems considering previous models suggested in the literature, because this is a key issue in the performance of the model-based fault detection system.

Several models for the pitch subsystem, the drive train, the tower and the blade root moment dynamics, the aerodynamics and the electrical subsystem were reviewed in this thesis. After the analysis and comparison of the models there were proposed one for each subsystem, in order to perform the further structural analysis and model calibration.

There is a variety of models for the drive train dynamics but the chosen one was the two shaft model which is the most common structure of the drive train present in wind turbine systems.

For the pitch subsystem different approaches were considered, some modeled this subsystem as a first order model and others a second order model. The second order models consider an hydraulic pitch system, which was the one used in the wind turbine reference in FAST and therefore it was the chosen model.

All the tower models found in the literature considered the movement of the structure in only one of the coordinates of the system, the fore-aft movement. Even though that in benchmark model and in the FAST simulator the movement of the tower in both directions (fore-aft and sideways), were considered. The simplification proposed for this subsystem is to consider it as a mass-spring system.

The aerodynamic models are quite complex, since they involve complex non-linear dynamics that are necessary to represent the behavior of the wind turbine and they describe how the power from the wind is extracted. Aerodynamics are present in the drive train dynamics and affect the tower structure of the wind turbine.

Blade root moment analytical models were not easy to find because in practice when it is wanted to study this dynamic, finite-element and other approaches are used to simulate and study this dynamic behavior. Even though the difficulties and scarce references it was possible to find an adequate analytical model that can be used for the intended purposes of this thesis.

The electrical subsystem models were quite standard in the literature, and are based on a first model for the generator/converter, being the ones proposed in this thesis.

Structural Analysis

Once the wind turbine subsystem models were proposed, the structural analysis based on the structural relations found in the analytical models was done. With the sensors available in the benchmark it was done

a matching procedure to estimate the unknown variables of the system from the known ones (sensors or measured variables). Once this is done, a set of analytical redundant relations was found to perform fault detection and isolation.

Calibrated Models

In order to implement the analytical redundant relations derived from structural analysis it is necessary to parameterize and adjust the models to the behavior observed from the sensor variables in FAST simulator. From the simulations, it was observed that some the drive train and blade root moment analytical redundant relations, needed to be redefined in order to approximate the behavior observed in the simulations. For the pitch subsystem, the generator/converter, and some of the drive train ARRs it was only necessary to do parameters estimations, which in several cases corresponded to the values of the nominal parameters taken from FAST 5-MW Wind turbine reference and the literature.

Fault Detection and Isolation

The model-based fault detection system was tested for the different fault scenarios proposed, demonstrating a good performance in the detection of the faults, leaving only one undetected fault due to to the threshold selection.

The fault isolation techniques FDI and DX applied to the signature matrix obtained from the tests, showed that only some of the faults were completely isolable. If a fault occurs in the pitch subsystem, it can be located the faulty subsystem but to isolate the fault between the actuator and the sensor is not possible with the current fault signature matrix.

6.1   Limitations and Future Work

The intention in the future and the objectives of the research project in which this master thesis was developed is to apply the fault detection system on a real wind turbine. To do that further research is need to be done towards the unmodeled phenomena present in some of the dynamics, for example the blade root moment dynamics.

The performance of the fault detection system was tested only assuming a constant wind speed. In real wind turbine systems the wind speed is constantly changing and presents turbulence. A more realistic approach needs to take into account this phenomena.

As explained in the introduction of this master thesis in future work the objective is not only to detect the fault, is also to do prognosis. This implies the ability to predict when the system is going to fail and estimate the remaining useful life of the wind turbine.

Bibliography

[1] Mid and long range plan for renewable energy development. 2007.

[2] 20% wind energy by 2030, increasing wind energys contribution to u.s. electricity supply. 2008.

[3] P. Seiler A.A. Ozdemir and G.J. Balas. Wind turbine fault detection using counter-based residual thresholding. 2011.

[4] Eduardo Quiles Cucarella Carlos Agudelo, Francisco Morant Anglada and Emilio Garca Moreno. Integration of techniques for early fault detection and diagnosis for improving process safety: Application to a fluid catalytic cracking refinery process. 2013.

[5] Pep Lluis Negre Carrasco. Fault detection and isolation of wind turbines - a real field data approach. 2010.

[6] Daus S. Vikatos T. Chantler, M.J. and Coghill G.M. The use of quantitative dynamic models and dependency recording engines. 1996.

[7] P. Dague M. dumas F. Levy J. Montmain ;. Staroswiecki Cordier, M. and L. Trav-Massuys. A comparative analysis of ai in control ttheory approaches to model-based diagnosis. 2000.

[8] C. Dobrila and R. Stefansen. Fault tolerant wind turbine control. 2007.

[9] Stijn Donders. Fault detection and identification for wind turbine systems, a closed-loop analysis. 2002.

[10] A. Wright E. Bossanyi and P. Fleming. Controller field tests on the nrel cart2 turbine. 2009.

[11] Thomas Esbensen and Christopher Sloth. Fault diagnosis and fault-tolerant control of wind turbines, 2009.

[12] Teresa Escobet. Structural analysis and analyticial redundancy. Universitat Politecnica de Catalunya, V International School on Fault Detection and Diagnosis of Complex Systems Madrid, Spain, July 2013.

[13] Hernan De Battista Fernando D. Bianchi and Ricardo J. Mantz. *Wind Turbine Control Systems. Principles,Modelling and Gain Scheduling Design.* 2007.

[14] L. Console Hamscher W. and J. de Kleer. Readings in model based diagnosis. 1992.

[15] C. Hatch. Improved wind turbine condition monitoring using acceleration enveloping. 2004.

[16] 2006 How Stuff Works. Horizontal-axis Turbine. $http : //static.howstuffworks.com/gif/wind - power - horizontal.gif$. 2006.

[17] $http : //environment.nationalgeographic.com/environment/global - warming/wind - powerinteractive.html$. National Geographic. Harness the power of wind.

[18] $http : //en.wikipedia.org/wiki/Wind_turbine$.

[19] Izadi-Zamanabadi. Fault-tolerant supervisory control - system analysis and logic design., 1999.

[20] Marshall L. Buhl Jr. J. Jonkman. *Fast User's Guide*, 2005.

91

[21] W. Musial J. Jonkman, S. Butterfield and G. Scott. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*, 2009.

[22] Juli Romera Joaquim Blesa, Vicenc Puig and Jordi Saludes. Fault diagnosis of wind turbines using a set-membership approach. 2011.

[23] K.E. Johnson and P. Fleming. Development, implementation, and testing of fault detection and condition monitoring on the national wind technology centers controls advanced research turbines. 2011.

[24] L.Y.and Balas K. Johnson, M. Pao and L. Fingeresh. Control of variable-speed wind turbines - standard and adaptive techniques for maximizing energy capture. 2006.

[25] F. Spinato M. R. Wilkinson and P. J. Tavner. Condition monitoring of generators and other subassemblies in wind turbine drive trains. 2007.

[26] M. Knowles M. R. Wilkinson, F. Spianto and P. J. Tavner. Towards the zero maintenance wind turbine. 2006.

[27] Emilio Garca Francisco Morant Mara Jos Prez, Antonio Correcher and Eduardo Quiles. Self-growing colored petri net for offshore wind turbines maintenance systems. 2011.

[28] H. Markou, T. Buhl, B.Marrant, and T.G van Engelen. *Morphological Study of Aeroelastic Control Concepts for Wind Turbines*, 2002.

[29] D. McMillan and G. W. Ault. Quantification of condition monitoring benefit for offshore wind turbines. 2007.

[30] Jan Lunze Mogens Blanke, Michel Kinnaert and Marcel Staroswiecki. *Diagnosis and Fault-Tolerant Control.* 2006.

[31] N. Sheibat-Othman N. Laouti and S. Othman. Support vector machines for fault detection in wind turbines. 2011.

[32] Peter F. Odgaard and Kathryn E. Johnson. Wind turbine fault detection and fault tolerant control - a second challenge. 2012.

[33] Peter F. Odgaard and Jakob Stoustrup. Results of a wind turbine fdi competition. 2012.

[34] The Encyclopedia of Alternative Energy and Sustainable Living. Wind Turbine. $http$: $//www.daviddarling.info/encyclopedia/w/ae_wind_turbine.html$. 2005.

[35] R. Nielsen P. F. Odgaard, J. Stoustrup and C. Damgaard. Observer based detection of sensor faults in wind turbines. 2009.

[36] D. Theilliol P. Poure, P. Weber and S. Saadate. Fault-tolerant power electronic converters: Reliability analysis of active power filter. 2007.

[37] Jakob Stoustrup Peter F. Odgaard and Michel Kinnaert. Fault tolerant control of wind turbines - a benchmark model. 2009.

[38] K. L. Sullivan J. F. Manwell R. W. Hyers, J. G. McGowan and B. C. Syrett. Condition monitoring and prognosis of utility scale wind turbines. 2006.

[39] Strategic research agenda: market deployment strategy from 2008 to 2030. European wind energy technology platform, july 2008. 2008.

[40] C. Svard and M. Nyberg. Automated design of an fdi system for the wind turbine benchmark. 2011.

[41] Michael Odgaard Niss Thomas Esbensen, Christoffer Eg Sloth and Brian Thorarins Jensen. Joint power and speed control of wind turbines. 2008.

[42] David Sharpe Tony Burton, Nick Jenkins and Ervin Bossanyi. *Wind Energy Handbook*, 2011.

[43] Teresa Escobet Vicenc Puig, Joseba Quevedo and Belarmino Pulido. On the integration of fault detection and isolation in model based fault diagnosis. 2004.

[44] A.H.A. Sari A. Naik A.Q. Khan W. Chen, S.X. Ding and S. Yin. Observer-based fdi schemes for wind turbine benchmark. 2011.

[45] C.A. Walford. Wind turbine reliability: understanding and minimizing wind turbine operation and maintenance costs. 200.

[46] Xiukun Wei and Michel Verhaegen. Fault detection of large scale wind turbine systems: A mixed h_∞/h_- index observer approach. 2008.

[47] M. Verhaegen X. Wei and T. van den Engelen. Sensor fault diagnosis of wind turbines for fault tolerant. 2008.

[48] S. Zhao R. M.G. Ferrari M. M. Polycarpou X. Zhang, Q. Zhang and T. Parisini. Fault detection and isolation of the wind turbine benchmark: An estimation-based approach. 2011.

[49] B. Bensaker Y. Amirat, M. E. H. Benbouzid and R. Wamkeue. Condition monitoring and fault diagnosis in wind energy conversion systems: a review. 2007.

[50] Y. Choa S. Ahnb Z. Hammeed, Y. Honga and C. Song. Condition monitoring and fault detection of wind turbines and related algorithms: A review. 2009.

Appendix A

Fault Detection Algorithm

```matlab
function y = fcn(res_1, res_2, res_3, res_4, res_5, res_6, res_7, res_8,
res_9, res_10, res_11, res_12,Time)

%This block implements the fault detection for each residual in every time
%instant and saves the alarm vector generated

%%%%%%%%%%%%%%%%%%%%%%%%%%
%No noise in the measures%
%%%%%%%%%%%%%%%%%%%%%%%%%%

    res_1_alarm=0;
    res_2_alarm=0;
    res_3_alarm=0;
    res_4_alarm=0;
    res_5_alarm=0;
    res_6_alarm=0;
    res_7_alarm=0;
    res_8_alarm=0;
    res_9_alarm=0;
    res_10_alarm=0;
    res_11_alarm=0;


if ((res_1 == 0) && (Time > 30))
    res_1_alarm = 1;
end


if ((res_2 == 0) && (Time > 200))
    res_2_alarm = 1;
end


if ((res_3 == 0) && (Time > 200))
    res_3_alarm = 1;
end

if ((res_4 == 0) && (Time > 200))
    res_4_alarm = 1;
end


if ((res_5 == 0) && (Time > 30))
    res_5_alarm = 1;
end


if ((res_6 == 0) && (Time > 20))
    res_6_alarm = 1;
end
```

```matlab
    if ((res_7 == 0) && (Time > 20))
        res_7_alarm = 1;
    end


    if ((res_8 == 0) && (Time > 20))
        res_8_alarm = 1;
    end


    if ((res_9 == 0) && (Time > 30))
        res_9_alarm = 1;
    end


    if ((res_10 == 0) && (Time > 20))
        res_10_alarm = 1;
    end

    if ((res_11 == 0) && (Time > 15))
        res_11_alarm = 1;
    end


    if ((res_12 == 0) && (Time > 25))
        res_12_alarm = 1;
    end

    alarms=[res_1_alarm res_2_alarm res_3_alarm res_4_alarm res_5_alarm res_6_alarm
        res_7_alarm res_8_alarm res_9_alarm res_10_alarm res_11_alarm res_12_alarm];
    y = alarms;
```

Published with MATLAB® 7.14

Appendix B

Blade Root Moment Mean Model

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%BRM estimation Test Mobile Means %
%Zone 1. Wind Speeds >= 12 m/s    %
%Mean(BRM)= a*mean(beta)          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clear all
load HH_24_arr.mat

%Pitch Angles Input Vector
X=[3.9064 6.6071 8.6733 10.4503 12.0524 13.5320 14.9157 16.2203 17.4687 18.6767
        19.8444 20.9696 22.0561 23.1117 24.1405 25.1451 26.1265 27.0848 28.0216]';

%BRM 1
Y1=[8392.2 7288.9 6553.8 5986.9 5521.1 5397.6 5024.5 4486.2 4219.3 3980.6 3762.5
3560.9 3373.3 3201.7 3040.3 2891.2 2754.7 2622.8 2499.9]';

P1_1=polyfit(X,Y1,1);
P2_1=polyfit(X,Y1,2);
P3_1=polyfit(X,Y1,3);

Ys1_1=P1_1(1)*X+P1_1(2);
Ys2_1=P2_1(1)*X.^2+P2_1(2)*X+P2_1(3);
Ys3_1=P3_1(1)*X.^3+P3_1(2)*X.^2+P3_1(3)*X+P3_1(4);

%Absolute Error
err1_1=sum(abs(Y1-Ys1_1))
err2_1=sum(abs(Y1-Ys2_1))
err3_1=sum(abs(Y1-Ys3_1))

%Mean Error
errm1_1=sum(abs(Y1-Ys1_1))/length(X)
errm2_1=sum(abs(Y1-Ys2_1))/length(X)
errm3_1=sum(abs(Y1-Ys3_1))/length(X)

%Quadratic Mean
ecm1_1=sqrt(sum((abs(Y1-Ys1_1))))/length(X)
ecm2_1=sqrt(sum((abs(Y1-Ys2_1))))/length(X)
ecm3_1=sqrt(sum((abs(Y1-Ys3_1))))/length(X)

%BRM 2
Y2=[8391 7290.1 6550.9 5986 5520.8 5398.9 5024.9 4486 4219.5 3977.8 3763.6
3559.7 3373.9 3202.3 3042.3 2891.7 2753.9 2622.1 2501.5]';

P1_2=polyfit(X,Y2,1);
P2_2=polyfit(X,Y2,2);
P3_2=polyfit(X,Y2,3);
```

```matlab
Ys1_2=P1_2(1)*X+P1_2(2);
Ys2_2=P2_2(1)*X.^2+P2_2(2)*X+P2_2(3);
Ys3_2=P3_2(1)*X.^3+P3_2(2)*X.^2+P3_2(3)*X+P3_2(4);


%Absolute Error
err1_2=sum(abs(Y2-Ys1_2))
err2_2=sum(abs(Y2-Ys2_2))
err3_2=sum(abs(Y2-Ys3_2))

%Mean Error
errm1_1=err1_2/length(X)
errm2_2=err2_2/length(X)
errm3_2=err3_2/length(X)

%Quadratic Mean
ecm1_2=sqrt(sum((abs(Y2-Ys1_2))))/length(X)
ecm2_2=sqrt(sum((abs(Y2-Ys2_2))))/length(X)
ecm3_2=sqrt(sum((abs(Y2-Ys3_2))))/length(X)


%BRM 3
Y3=[8392 7289.5 6553 5986.8 5521.2 5396.9 5021.1 4484.6 4218.1 3979.1 3759.5

3559 3373.5 3201.7 3040.3 2891.9 2751.9 2621.2 2500]';


P1_3=polyfit(X,Y3,1);
P2_3=polyfit(X,Y3,2);
P3_3=polyfit(X,Y3,3);


Ys1_3=P1_3(1)*X+P1_3(2);
Ys2_3=P2_3(1)*X.^2+P2_3(2)*X+P2_3(3);
Ys3_3=P3_3(1)*X.^3+P3_3(2)*X.^2+P3_3(3)*X+P3_3(4);


%Absolute Error
err1_3=sum(abs(Y3-Ys1_3))
err2_3=sum(abs(Y3-Ys2_3))
err3_3=sum(abs(Y3-Ys3_3))

%Mean Error
errm1_3=err1_3/length(X)
errm2_3=err2_3/length(X)
errm3_3=err3_3/length(X)


%Quadratic Mean
ecm1_3=sqrt(sum((abs(Y3-Ys1_3))))/length(X)
ecm2_3=sqrt(sum((abs(Y3-Ys2_3))))/length(X)
ecm3_3=sqrt(sum((abs(Y3-Ys3_3))))/length(X)
```

```matlab
%BRM Average
Yavg=[8.3896e+03 7.2895e+03 6.5527e+03 5.9856e+03 5.5212e+03 5.1281e+03 4.7876e+03
 4.4882e+03 4.2221e+03 3.9836e+03 3.7666e+03 3.5662e+03 3.3804e+03 3.2079e+03
     3.0478e+03 2.8986e+03 2.7593e+03 2.6290e+03 2.5068e+03]';
P1_avg=polyfit(X,Yavg,1);
P2_avg=polyfit(X,Yavg,2);


Ys1_avg=P1_avg(1)*X+P1_avg(2);
Ys2_avg=P2_avg(1)*X.^2+P2_avg(2)*X+P2_avg(3);

BRM1_err1_avg=sum(abs(Y1-Ys1_avg));
BRM1_err2_avg=sum(abs(Y1-Ys2_avg));



BRM2_err1_avg=sum(abs(Y2-Ys1_avg));
BRM2_err2_avg=sum(abs(Y2-Ys2_avg));


BRM3_err1_avg=sum(abs(Y3-Ys1_avg));
BRM3_err2_avg=sum(abs(Y3-Ys2_avg));

%BRM Mean Model Comparison. Estimated and Simulated

BRM1_est=P2_1(1)*Beta_1(:,2).^2+P2_1(2)*Beta_1(:,2)+P2_1(3);
BRM2_est=P2_2(1)*Beta_2(:,2).^2+P2_2(2)*Beta_2(:,2)+P2_2(3);
BRM3_est=P2_3(1)*Beta_3(:,2).^2+P2_3(2)*Beta_3(:,2)+P2_2(3);


BRM1_3rd=P3_1(1)*Beta_1(:,2).^3+P3_1(2)*Beta_1(:,2).^2+P3_1(3)*Beta_1(:,2)+P3_1(4);
BRM2_3rd=P3_2(1)*Beta_2(:,2).^3+P3_2(2)*Beta_2(:,2).^2+P3_2(3)*Beta_2(:,2)+P3_2(4);
BRM3_3rd=P3_3(1)*Beta_3(:,2).^3+P3_3(2)*Beta_3(:,2).^2+P3_3(3)*Beta_3(:,2)+P3_3(4);
```

Published with MATLAB® 7.14

Appendix C

Residuals Simulink Implementation

In this appendix is showed and explained how the residuals of the fault detection system illustrated in Figure 5.4 were implemented. The residuals implemented use all the calibrated models explained in Chapter 4. The residuals have as inputs the signals from the sensors and as outputs an alarm vector indicating that the threshold limits were violated.

Residual 1. Power Subsystem The residual implementation of the power subsystem is showed on figure C.1. This figures corresponds to the implementation of the analytical redundant relation (3.4) calibrated in section (4.5).
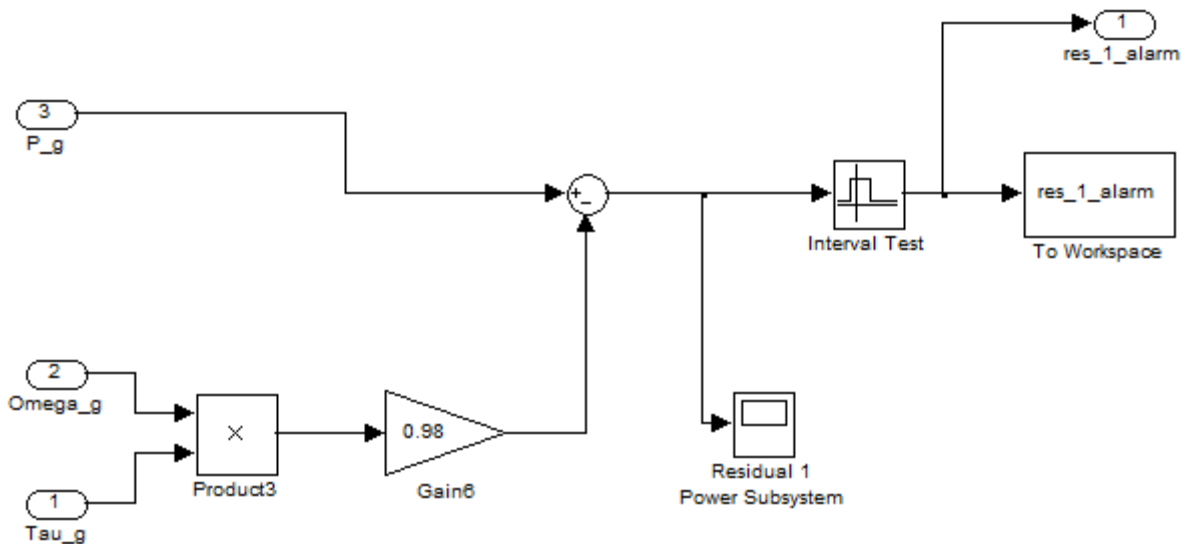
Figure C.1: Power subsystem residual implementation

Residual 9. Drive Train Subsystem The residual 9 implementation for the drive train subsystem with the variables generator speed and torque is showed on Figure C.2. This implementation is based on the subsystem model calibrated in Section 4.7.
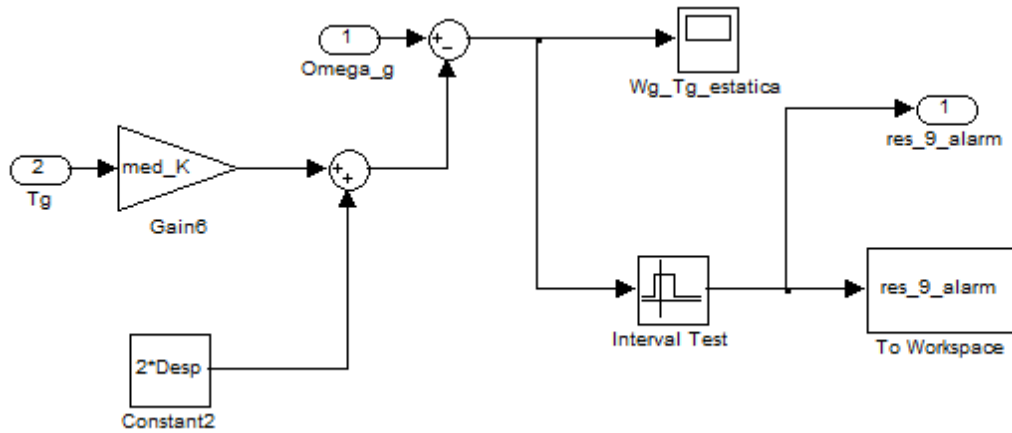
Figure C.2: Residual 9 Drive Train Subsystem

Residual 12. Drive Train Subsystem. The residual 12 implementation for the drive train subsystem derived from the state space modified model with rotor speed as output variable is shown on Figure C.3.
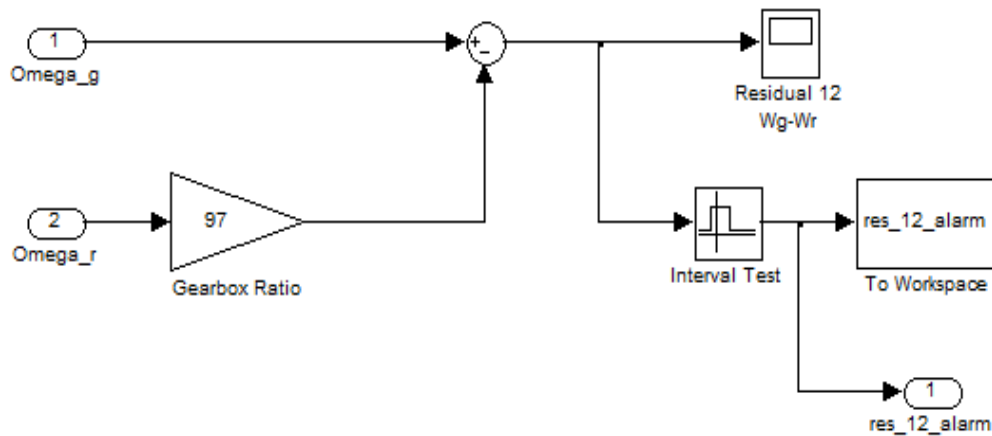

Figure C.3: Residual 12 Drive Train Subsystem

Residuals 2,3,4. Blade Root Moments of the three wind turbine blades. These residuals implementation corresponds to the blade root moments mean models calibrated in section 4.6.1. The blade root moment residual for blade 1 is showed on Figure C.4. It is the same structure for blade root moments of blades 2 and 3.

ssdt8

In the implementation of the mean blade root moment proposed model it was necessary to include a saturation for the values of the initial time period in order to obtain the closest value of the estimated mean blade root moment signal in steady state regime. The mean model was estimated in the steady state regime when the signal started to oscillate inside a limited band, as shown in Figure **??**.
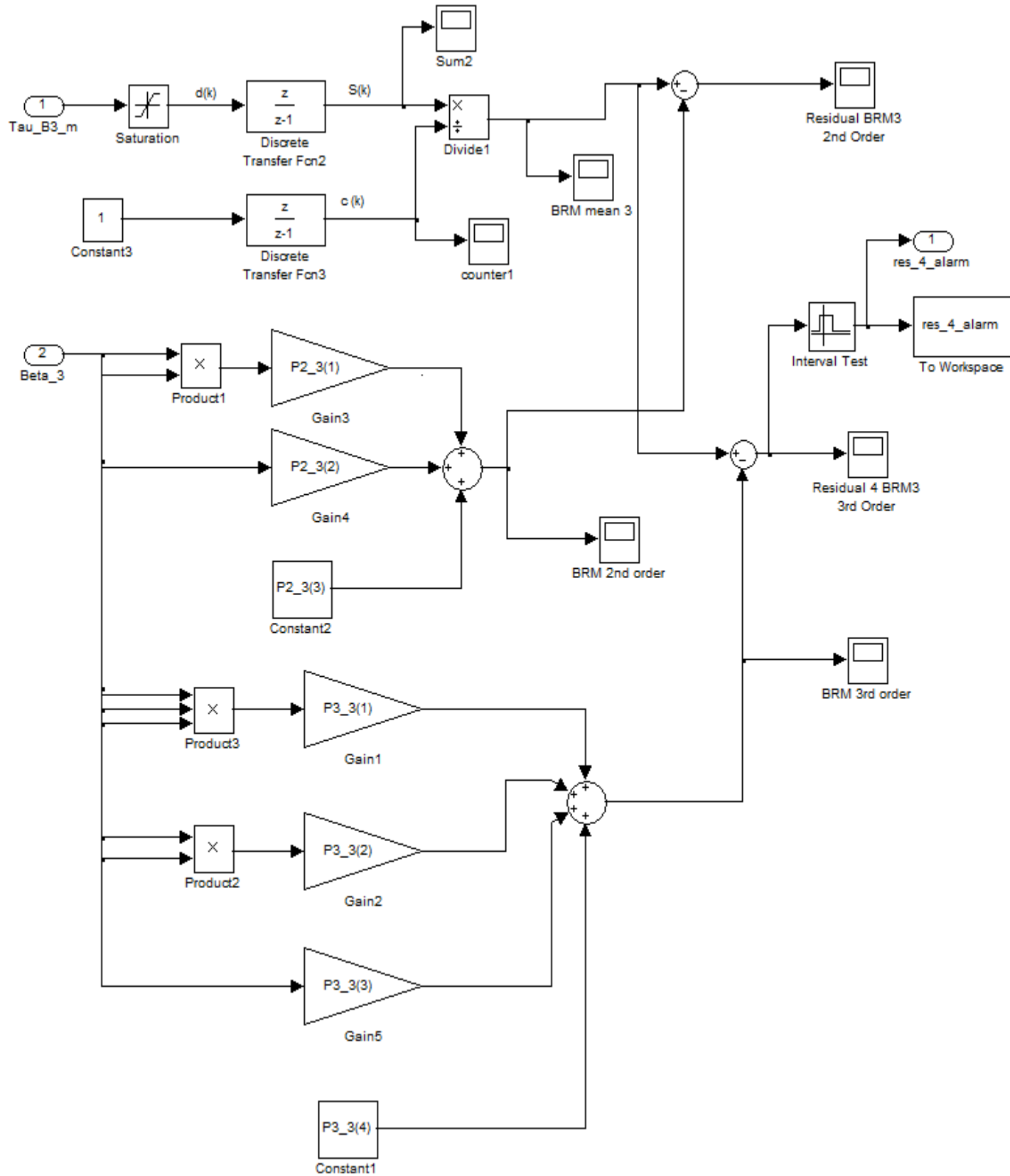
Figure C.4: Blade Root Moment 1 residual implementation

Residual 5. Generator and Converter Subsystem. The residual implementation for the generator and converter subsystem is shown in Figure C.5. This implementation is based on the subsystem model calibrated in Section 4.3.
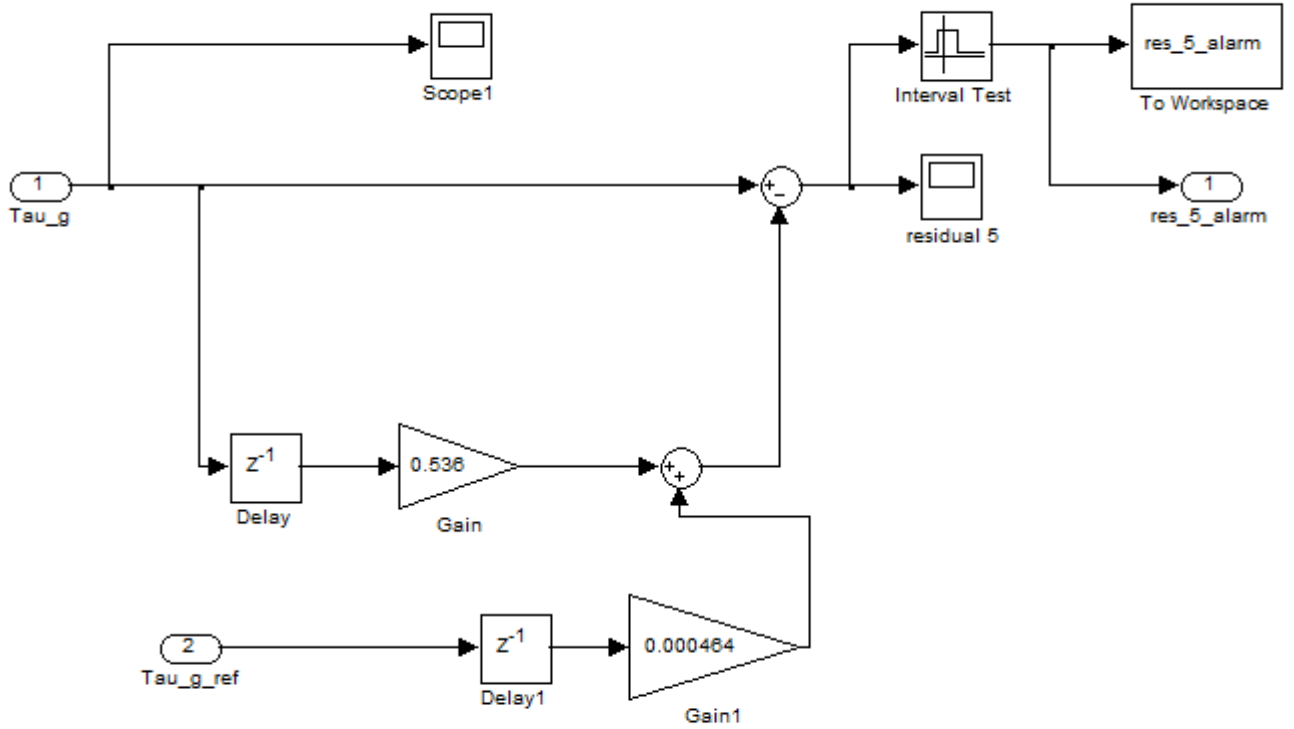

Figure C.5: Residual 5 Generator and Converter Subsystem

Residual 6. Pitch Subsystem

The residual implementation for the pitch subsystem 1 is shown in Figure C.6. It is the same structure for pitch subsystems 2 and 3. This implementation corresponds to the calibrated pitch subsystem detailed in Section (4.4.
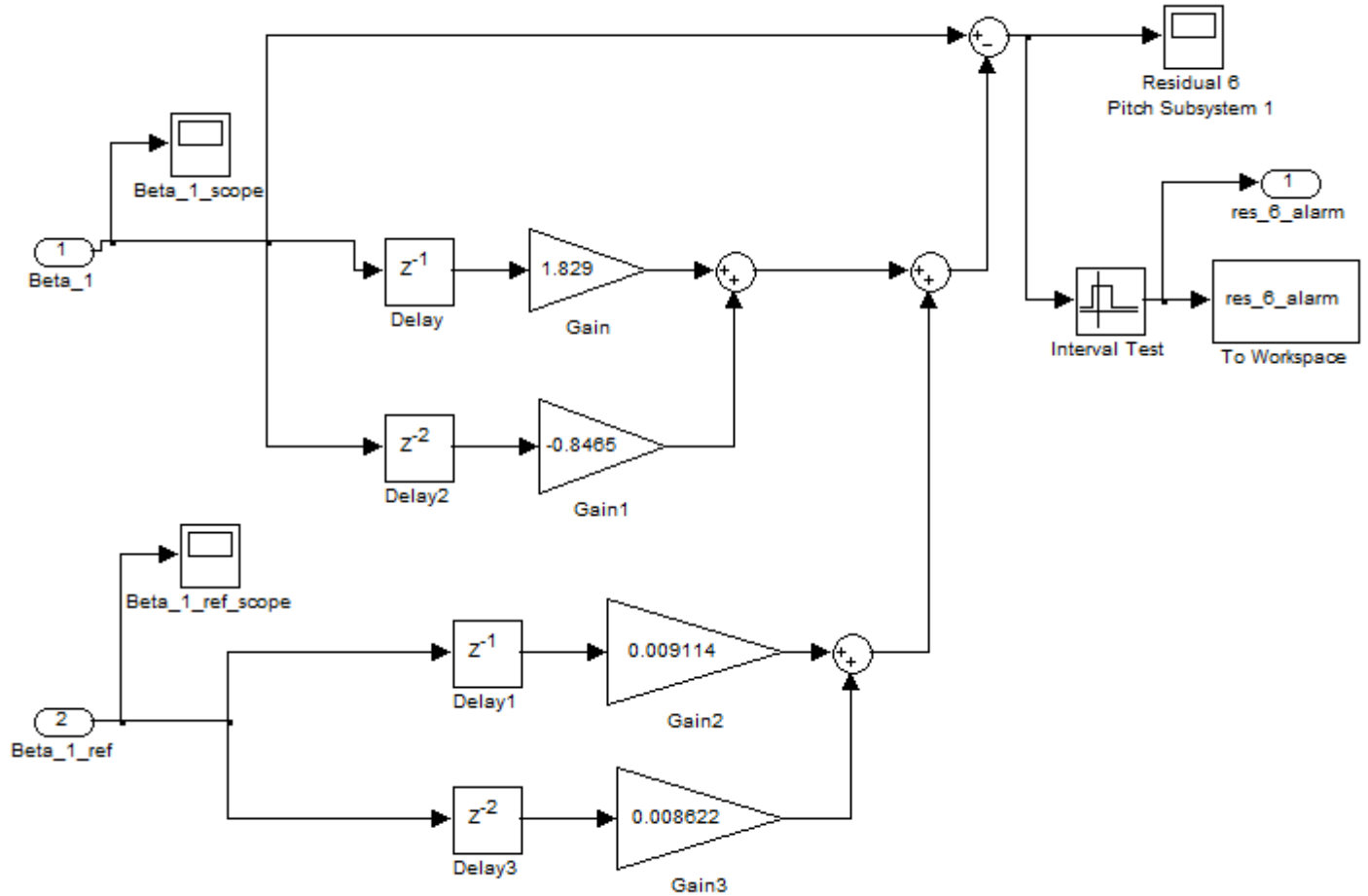

Figure C.6: Residual 6 Pitch Subsystem

Residual 10. Drive Train Subsystem. The residual 10 implementation for the drive train subsystem with the variables generator speed and torque is based on the calibrated model showed in Section 4.7. The implementation scheme is shown in Figure C.7.
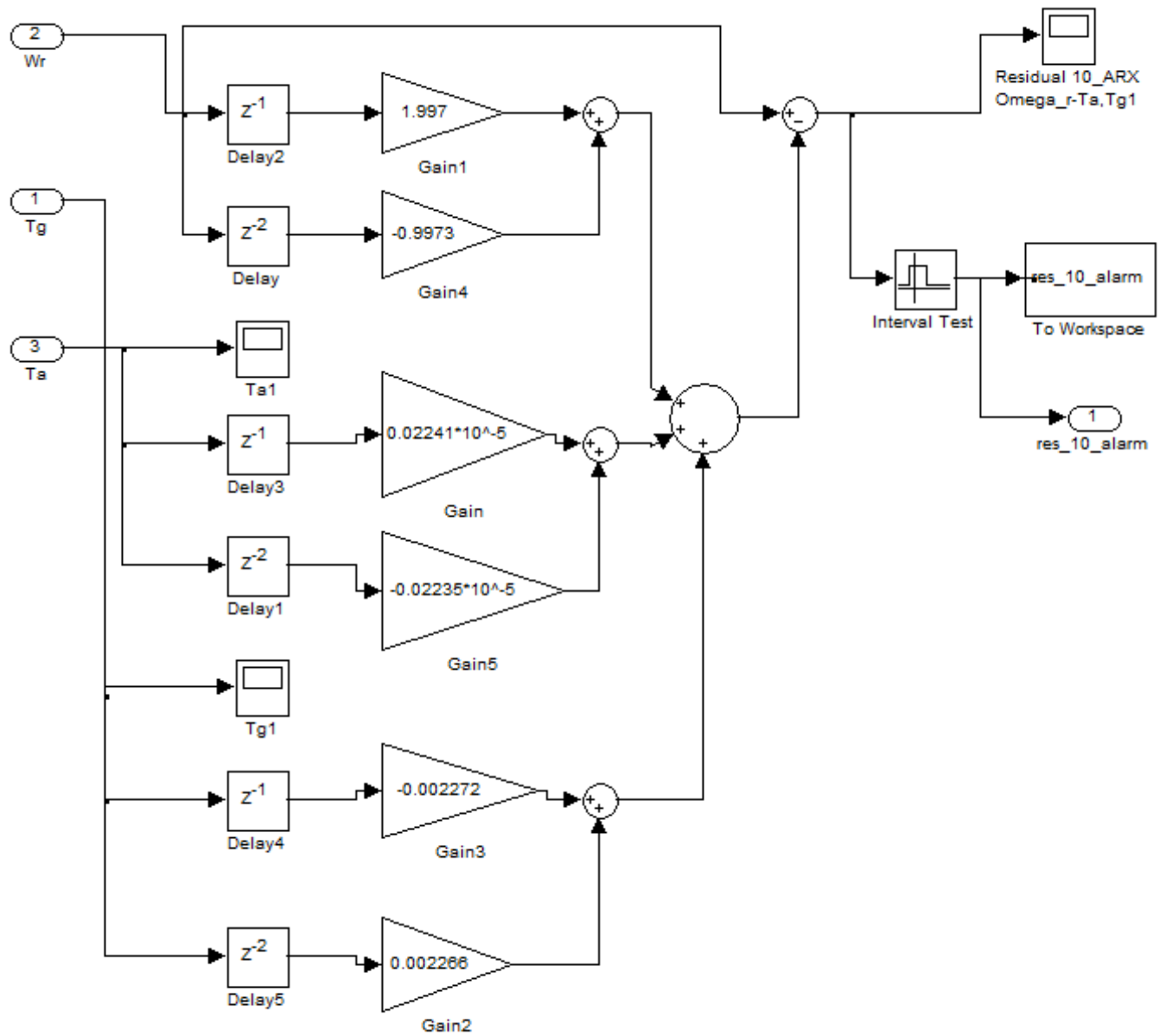

Figure C.7: Residual 10 Drive Train Subsystem

Residual 11. Drive Train Subsystem. The residual 11 implementation for the drive train subsystem derived from the state space modified model with generator speed as output variable showed in Section 4.7. The implementation scheme is shown in Figure C.8.
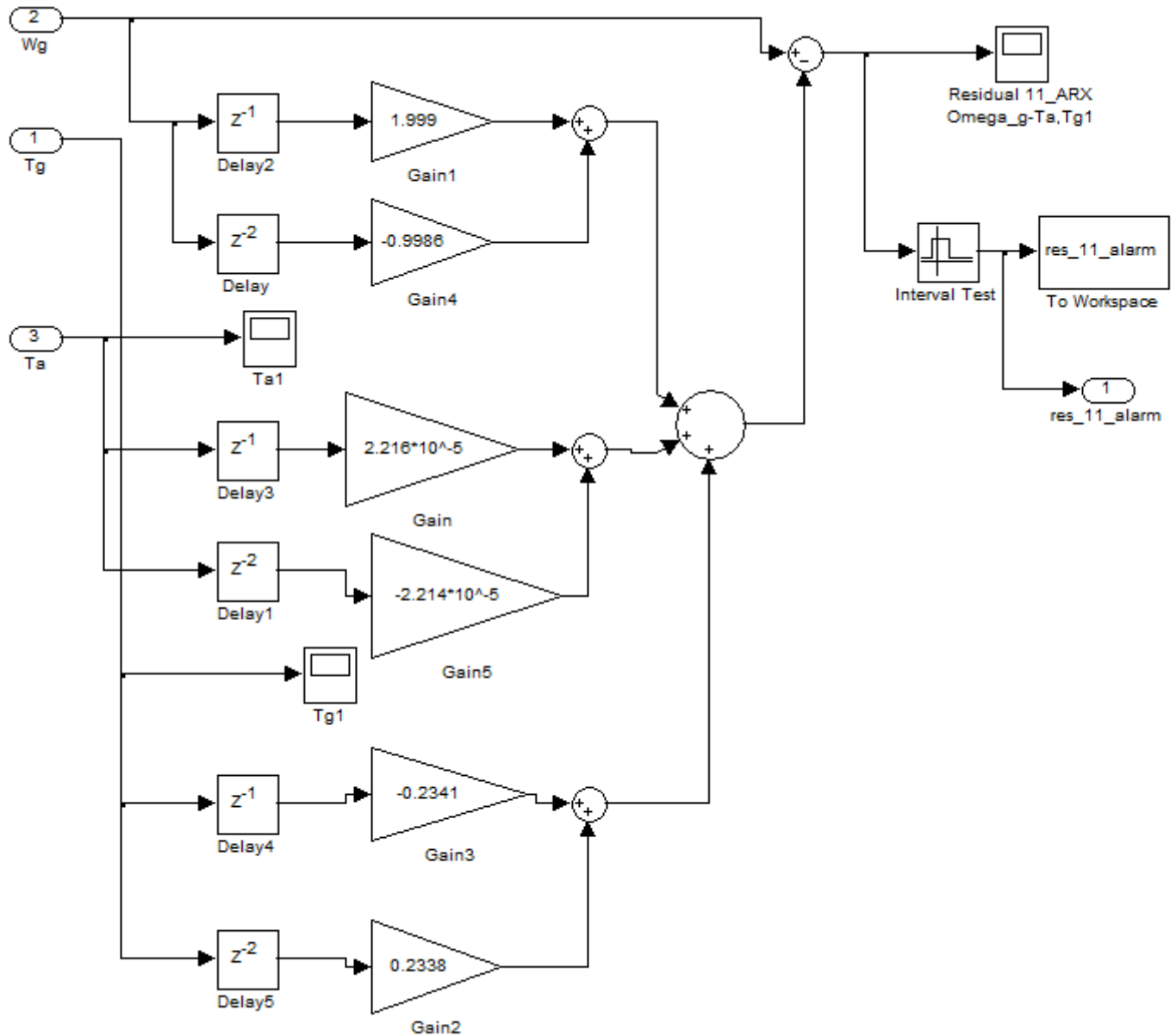

Figure C.8: Residual 11 Drive Train Subsystem