FAULT DIAGNOSIS AND FAULT TOLERANT CONTROL FOR WIND TURBINE DYNAMIC SYSTEMS

KINGSLEY MADUBUIKE

A thesis submitted in partial fulfilment of the requirements of the Liverpool John Moores University for the degree of Doctor of Philosophy

MARCH 2022

ABSTRACT

The aim of this research is to develop a fault tolerant control (FTC) and fault diagnosis (FD) methodology for nonlinear dynamic systems. This method is applied to the pitch system of a variable speed wind turbine system to verify the effectiveness of this method. The research is divided into three parts.

The first part proposes a robust fault detection approach using an unknown input observer method. This developed observer is sensitive to actuator faults in the benchmark model while it is robust to the system disturbance. A benchmark model consisting of a pitch system, a drive train, generator and converter and the state space model is proposed.

The second part proposes the use of neural network (NN) estimator to detect sensor faults in a wind turbine system. An independent radial basis function neural network (RBFNN) is developed for online diagnosis of the sensor faults. The RBF is trained using sample data during a fault free operating condition. The benchmark model of the wind turbine system proposed in the first part with three sensor faults simulated is developed on Simulink. This research will use two techniques to employ the RBF. One of the RBF will be used to model the wind turbine and generate residuals while the second RBF is developed as a classifier to isolate faults from the generated residuals.

In the final part the reliability of the wind turbine system is guaranteed by designing a variable speed wind turbine pitch angle control which can tolerate and detect faults. The pitch angle system consists of hydraulic pump, hydraulic and pitch gear system. The fault diagnosed here is the shaft rotary friction change which is caused by the break of shift or pitch gear set. The proposed fault tolerant control (FTC) method uses a disturbance observer to diagnose the fault. The FTC is implemented using the combination of a neural network (NN) estimator and a full order terminal sliding mode control. The post fault states can drive to the sliding surface and converge in finite time. The control law is derived using a Lyapunov stability method to ensure we guarantee stability for post-fault. Matlab and Simulink are used to simulate the electrohydraulic servo pitch system with the faults simulated. A third order nonlinear state space is derived, and physical parameters applied in the simulation. The simulation results are presented to validate the effectiveness of the proposed method.

ACKNOWLEDGEMENT

I would like to express my utmost gratitude to God almighty for the grace to complete my research studies.

I would like to express my gratitude to Prof Dingli Yu for his continuous support all through my PhD study and research, and for his motivation and patience. This work would not be possible without his invaluable suggestions and guidance. I appreciate all your effort to help me expand my knowledge through self-learning. I could not have imagined having a better supervisor and mentor for my PhD study. I would also like to appreciate my co supervisors' Dr Barry Gomm and Dr Qian Zhang for their contributions during my PhD study.

I would like to express my appreciation to my wife Netochukwu Madubuike and my son King David Madubuike for their endless love and support during this thesis. I wish to express my deepest gratitude to my parents Engr and Mrs L N S Madubuike for their love and support during my PhD study. I also want to express my appreciation to my brothers and sisters for their undivided support all through my PhD study.

I would also wish to acknowledge the contribution of Engr Oghenefe Otobrise in proofreading every chapter of this thesis.

I also want to acknowledge the effort of the faculty of engineering and technology and the Department of Electrical and Electronics Engineering in making sure that the Mat lab/ Simulink software was made available.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Variables	Definitions
$\boldsymbol{u(t)}$	Servo motor control
$\boldsymbol{w_p(t)}$	Rotary speed of the servo motor
$\boldsymbol{q_1(t)}$	Inlet flow rate of hydraulic pump
$\boldsymbol{q_2(t)}$	Outlet flow rate of hydraulic pump
$\boldsymbol{P_1(t)}$	Chamber pressure of hydraulic motor in high pressure side
$\boldsymbol{P_2(t)}$	Chamber pressure of hydraulic motor in low pressure side
$\boldsymbol{q_{r1}(t)}$	Flow rate through inlet relieve valve
$\boldsymbol{q_{r2}(t)}$	Flow rate through outlet relieve valve
$\boldsymbol{w_m(t)}$	Rotary speed of the hydraulic motor
$\boldsymbol{\theta_m(t)}$	Rotary displacement of hydraulic motor
$\boldsymbol{f(t)}$	Load torque on the motor shaft
$\boldsymbol{\beta(t)}$	Pitch angle
$\boldsymbol{T_p(t)}$	Added viscous and column friction of pitch angle system axis and gear set caused by occurred component failure.
$\boldsymbol{K_a}$	Gain of the servo motor unit
$\boldsymbol{D_p}$	Volumetric displacement of the pump
$\boldsymbol{C_{tp}}$	Leakage coefficient of the pump
$\boldsymbol{\beta_e}$	Bulk modulus of the hydraulic fluid

V_1	Original total control volume of the hydraulic motor chamber in the high-pressure side.
V_2	Original total control volume of the hydraulic motor chamber in the low-pressure side.
D_m	Volumetric displacement of the motor
C_{tm}	Leakage coefficient of the motor
J_t	Total inertia of hydraulic motor
G	Pitch load spring gradient
i_g	Pitch angle gear ratio
A	Area covered by the rotor
R	Radius of the blade
ρ	Air density
v_w	Speed of the wind
P_w	Available power by rotor
C_p	Power coefficient
P_r	Rotor power
λ	Tip-speed ratio
β_{ref}	Reference blade pitch angle
$T_{g_{ref}}$	Reference generator torque
ω_r	Mechanical angular velocity
T_r	Rotor torque
ω_n	Natural frequency of pitch actuator

ζ	Damping ratio of pitch actuator
N	Gear ratio
J_r	Moment of inertia of rotor
J_g	Moment of inertia of generator
D_s	Driveshaft damping constant
K_s	Driveshaft spring constant
T_g	Generator torque
C_q	Torque coefficient
β_r	Desired pitch angle
$w(k)$	weight of the matrix
$h(k)$	The Output vector of the hidden layer
$\hat{y}$	the output vector of the RBF network
σ	is a positive scalar known as width
C_i	Centres
$y_c(k)$	is the process output vector
μ	is the forgetting factor
γ	is the inverse of a learning rate.

CHAPTER 1
Introduction

1.1 Background of thesis

Recently, wind energy has attracted more attention from energy suppliers around the world; this is due to increased government investments and promotion of going green. These have led to the demand for wind energy because it is renewable and eco-friendly [1][2]. Over the years, wind turbines have contributed significantly to power production around the world [3]. By the end of 2019, the wind turbine installed capacity worldwide had reached 650.8GW [106]. Wind turbines have had an average yearly increase rate of more than 10 percent over the past 10 years [106]. Megawatt-sized wind turbines are expensive; therefore, it is anticipated that their energy production within the short downtimes will be worth the investment [3]. In modern engineering systems, the demand for improved productivity leads to more challenging operating conditions [4]. Wind turbines are generally located in isolated places and some are made to work in difficult conditions [107][58]. Figure 1.0 presents the components of the wind turbine.

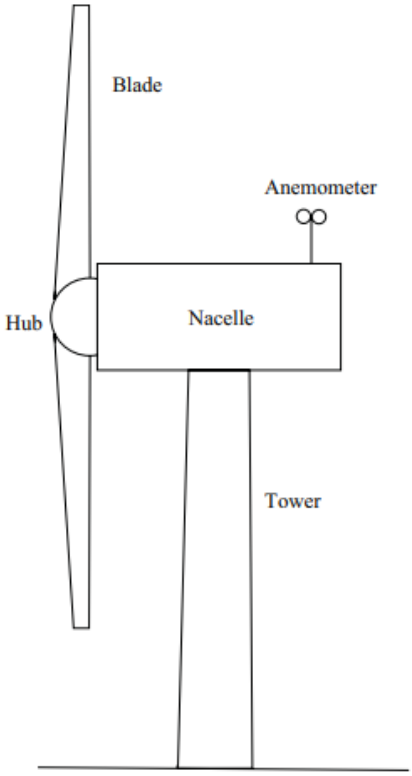

Figure 1.0: Components of the wind turbine [5][115]

This has made accessibility to these turbines difficult and costly if a fault occurs [107]. This has led to an increase in demand to improve the control system of the wind turbine with fault tolerant control schemes that allows the wind turbine to continue operating even when a fault occurs [107]. A wind turbine is a complex system but when some components fail in the wind turbine it can affect the entire wind turbine system [58]. From a commercial point of view, the introduction of fault tolerant control and fault diagnosis will reduce the maintenance cost and make wind turbine commercially more competitive especially when compared to other types of energy sources [58].

A fault in a wind turbine system can be defined as an unpermitted deviation of the structure of a system from the nominal situation. Faults are often unexpected and impossible to predict. In a wind turbine, the blade pitch system contributes to the failure rate and downtime of a wind turbine [108]. The gearbox faults are a major category of failure modes in wind turbine (WT) drive-train systems. They are caused by various factors such as manufacturing and installing errors, surface wear, fatigue, etc. A gear fault will lead to performance degradation of the WT drivetrain and may cause a catastrophic failure of the gearbox or even failures of other components in the WT drivetrain [114]. Faults can be classified as actuator, component, or sensor faults. The effect of faults in a system range from total shut down of the system to delay in production until the fault is resolved [3]. From an industrial point of view, different faults can occur in a wind turbine system. Pitch systems could have actuator faults. This actuator fault in the pitch may be caused by air content in the oil or hydraulic pressure drop. This will lead to slow pitch action which makes it impossible for the pitch control to maintain the rated speed in the rotor. The sensor fault can occur by damage to the sensor caused by the maintenance team. This will lead to a false measurement of the pitch angle, rotor speed and generator speed.

A fault tolerant control system should have the ability to detect faults early without human intervention and accommodate the effects of the faults on the system in such a way that the system can maintain working while the performance provided is acceptable [3][76]. A fault diagnosis system is a system that has the capacity of detecting and isolating faults [39]. The key for protecting the turbine from an unexpected operational condition is fault diagnosis of a blade pitch system at the early stage [108]. Different methods have been developed recently for fault tolerant control such as the use of adaptive neural network [116], terminal sliding mode control [119], back stepping [117], least squares support vector machine [118]. In the actual application, these methods have their own advantages and disadvantages [121]. The least squares support vector machine can resolve the nonlinear, small-sample, and high dimensional problems [121]. When least squares support vector

machine is used to solve nonlinear problems, the kernel function will affect the final classification result directly [121]. Robustness and fault tolerance capabilities are vital properties in the control system of a wind turbine system [22]. The wind turbine design has a couple of challenges, but the main challenges are:

(1) the aerodynamic torque is not measured [130].

(2) the wind speed is difficult to measure as the speed is not constant [130].

Due to competition between wind turbine producers, wind turbine data are not always available. These challenges led researchers to develop modelling techniques for the wind turbine systems.

Modelling has become the basic tool for analysing dynamic systems including wind turbines. The last two decades have recorded significant research activities in fault diagnosis and accommodation analysis. [7] talked about the fault detection and isolation of the wind turbine benchmark using an Estimation-based Approach. In [8], a robust detection and isolation scheme was developed for abrupt and incipient faults in nonlinear Systems. The model-based fault detection and the signal-based fault detection were compared, and the result presented.

1.2 Aims and Objectives

The aim of this project is to propose a novel fault detection and isolation (FDI) approach for nonlinear industrial plants and implement the proposed approach to a wind turbine system to evaluate the effectiveness of the developed approach. The following objectives are designed to achieve this aim.

(a) Develop a mathematical model for the wind turbine system including pitch system, drive train and generator parts.

(b) Design the wind turbine system on Matlab/Simulink.

(c) Develop fault diagnosis for wind turbine system using unknown input observer. Simulate the faults and analyse the result.

(d) Design a nonlinear state observer including a neural network estimator to diagnose the faults. Simulate three sensors faults and validate the results.

(e) Propose and design full order terminal sliding mode control with finite time stability, no singularity, no chattering features for fault diagnosis of the pitch system and drive train of the wind turbine system.

(f) Design, configure and model the pitch system on Matlab/Simulink.

(g) Evaluate the developed fault diagnosis and FTC methods on the pitch system developed using Matlab/Simulink.

1.3 Novelty

The novelty of this research is summarised as follows.

i. A novel method will be proposed to design the nonlinear observer and adaptation law for neural estimator to estimate the states which will enable identifications of actuator and sensor faults, as well as provide some further information about the faults.

ii. A novel terminal sliding mode control method will be developed for fault tolerant control of the pitch systems of the wind turbine. This design has the advantages of robustness, rapid response and converges at a finite time.

iii. The stability of the closed-loop system after the occurrence of a fault is guaranteed using Lyapunov theory.

iv. The pitch angle control is achieved in the presence of friction change which is a fault that often occurs in real life wind turbine systems.

v. The developed new control method will be applied to a simulation of the pitch systems of a wind turbine system for fault tolerant control, so that the control performance will be more reliable, robust and stabilized for post-fault dynamics

1.4 Structure of Thesis

This thesis is organised in seven chapters to achieve the objectives above.

In Chapter One, the background of the thesis has been presented. Wind energy has attracted more attention from energy suppliers around the world; this is due to increased government investments and promotion of going green. In modern engineering systems, the demand for improved productivity leads to more challenging operating conditions. This is hard to achieve without the introduction of fault detection, isolation, and accommodation systems into the turbines. The aim and objectives of this research are also presented.

In Chapter Two, cumulative research work has been carried out by several researchers over the past few decades. A fault diagnosis system is a system that has the capacity of detecting and isolating faults. An in-depth study of faults is presented. Different types of real time faults in a wind turbine are presented here. Fault detection types are presented and explained in detail. Fault

tolerant control is also presented here with the two types explained in this chapter. Fault detection using artificial intelligent is also discussed here.

In Chapter Three, the benchmark model including all subsystems of the wind turbine are modelled and described. The Simulink model of the nonlinear wind turbine including the pitch angle, aerodynamic, drive train, generators and converters are set up using Matlab/Simulink. The state observers are designed for the wind turbine system.

In Chapter Four, the fault detection and isolation for pitch and drive train model using unknown input observer is described. An unknown input observer is designed, and this observer is sensitive to faults and robust to system disturbances. An actuator fault and three sensor faults are simulated on Matlab/Simulink. The simulation performances are presented to show the effectiveness of the proposed method on the linear model of the wind turbine.

Chapter Five investigates the fault diagnosis for wind turbine systems using a neural network estimator. Firstly, an independent RBFNN is designed to estimate the states and generate residuals for the detection part. The training algorithm of RBF is outlined, and the data selection process is presented. Fault detection was carried out using RBF network. First at no fault condition for training purposes and then a fault is simulated, and the simulation result was as desired. For the fault isolation task, an additional RBFNN is designed as a classifier. The simulation results are presented.

In Chapter Six, the pitch system design, configuration and modelling of the hydraulic pitch system is presented. The variable and physical parameters of the model are presented.

Chapter Seven presents a fault detection and fault tolerant control scheme using the proposed method. For the fault detection scheme, a disturbance observer is designed, and an adaptive RBF is also presented. The proposed method used for fault tolerant control is modelled here. The proposed method was applied to the hydraulic pitch system in this chapter. The simulation results are presented and discussed.

In Chapter Eight, the general conclusions of this research described in the thesis are presented. Suggestions and recommendations on how the research can be further developed are also presented.

CHAPTER 2
Literature Review

2.1 Introduction

The wind turbine systems are nonlinear, unsteady, and complex [8]. Implementing maximal energy transformation and adaptation to wind speed change are important roles that the control system plays in wind turbines [8]. However, wind turbines are situated in a harsh atmosphere, which makes them prone to large numbers of accidents [20]. Due to the accidents, the need for operational reliability, safety and stability is rising. However, unplanned maintenance is expensive, which demonstrates the use of an advanced control system to balance load reduction and power generation optimally.

2.2 Fault diagnosis

With the rapid development of complex industrial systems, it is important to maintain increasing demands for reliability and safety of control systems [6]. Due to the large production requirements of modern engineering systems, there is need for more demanding operating conditions [7]. Some of these conditions increase the chances of faults occurring in the system. A fault diagnosis system is a system that has the capacity of detecting and isolating faults [39]. Fault diagnosis is subdivided into three.

- Fault detection
- Fault isolation and
- Fault accommodation.

I.Faults

In dynamic systems, a fault can be defined as an unpermitted deviation of a system structure from the nominal situation [8][40][73]. This means that a fault is a state that could lead to a system malfunctioning or failure [30][66]. There are three different types of faults from a control engineer point of view [8]. These are shown in Figure 2.0. Faults in dynamic systems are normally caused by malfunctioning of the actuators, sensors, or other components in the system [5][63]. When there are sensor faults, plants are not affected but the sensor readings have significant errors [8]. On the other hand, if there is an actuator fault, the plant properties are not affected but the impact of the controller on the plant is either distributed or modified [8]. In the case of plant also called component fault, the input and output properties of the plant are changed [8]. These faults may lead to unwanted

performance or cause instability and damage to the system, which may further lead to the shutdown of the system [5][6].

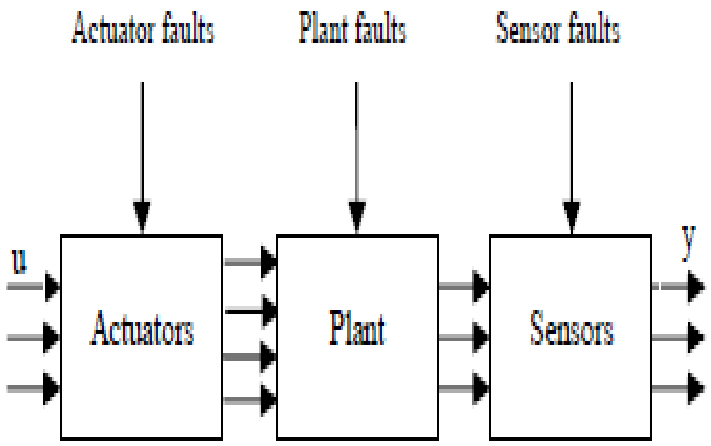

Figure 2.0: Different types of faults [8]

Changes in faults may be abrupt, gradual, or intermittent and their time behaviour may be unpredictable [40][89]. Due to the damage that faults may cause, engineers seek to design a reliable control system [5][9].

II.Fault diagnosis process

The process of fault diagnosis is made up of two phases: fault detection and fault isolation. This is the reason why many papers use FDI (fault detection and isolation). This fault diagnosis can be further achieved by the combination of fault generation and residual evaluation [40][35]. This can be presented in Figure 2.01.

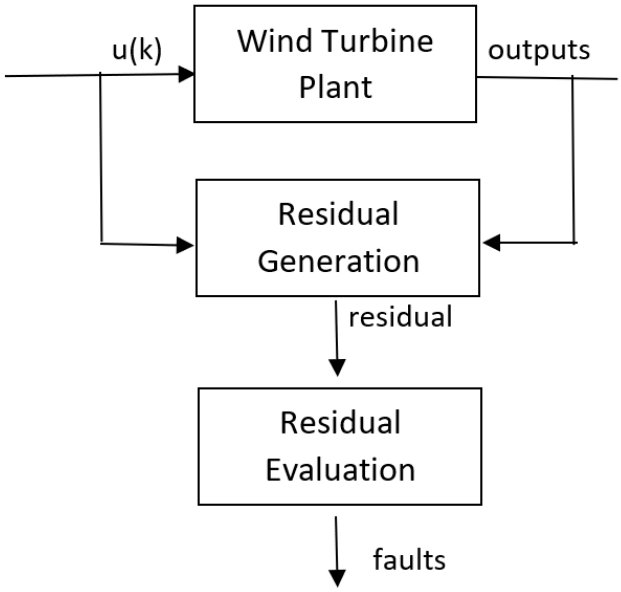

Figure 2.01: Fault diagnosis stages [35][40]

From Figure 2.01, the fault diagnosis process has two systems namely residual generation and residual evaluation [40].

i. Residual generation

The residual generation deals with comparing the measured system outputs and the predicted system outputs. The reason for this procedure is to calculate the quantitative indices of fault presence in the residuals [40][86]. For a fault free condition, the residual is expected to be close to zero but at the occurrence of a fault it is expected to deviate from zero [35][40].

ii. **Residual evaluation**

There is need for a process that translates or analyses the residual signal to determine if there is a fault [35]. This process is known as residual evaluation. This process determines whether a fault has occurred, and it helps in isolating the fault in the system device [40]. It also determines whether the fault is small or not [40].

2.2.1 Fault detection (FD)

The objective of fault detection is to determine the occurrence of a fault [8]. It decides the time it takes a system to be exposed to faults [8][11]. It can be determined with or without a process model [35]. The safety of real systems is guaranteed when faults are detected early [36].

2.2.1.1 Types of fault detection

FD is subdivided into four as presented in figure 2.02. These are further explained below.

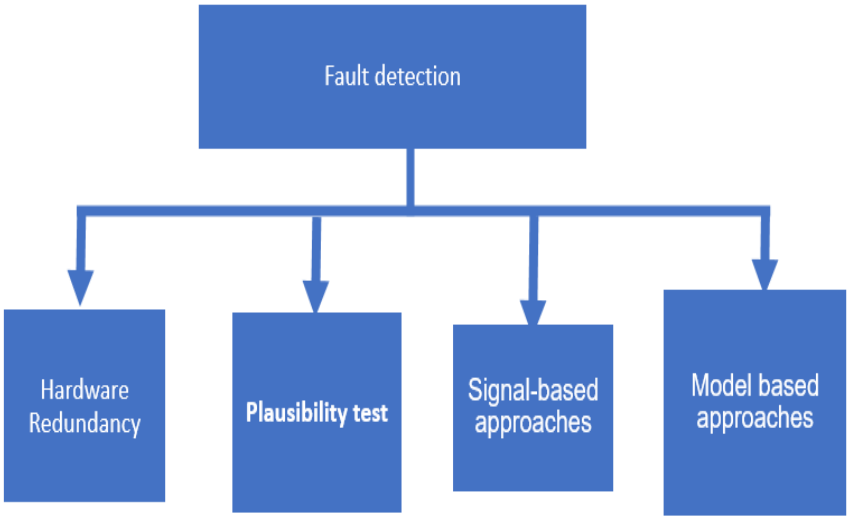

Figure 2.02: Types of fault detection

a. *Hardware redundancy:*

This is the traditional engineering approach for fault detection in dynamic systems [102]. Here, the process components are rebuilt using identical hardware components with the same input signals to produce duplicate output signals that can be compared [8][11]. If the output of the process component is different from the one of its redundancy, then there is a fault in the process component [8][40][11]. Hardware redundancy is dependable, but costly, which increases weight of systems and occupies more space [12]. This has made this approach unattractive.

b. *Plausibility test*

The plausibility test is carried out based on the check of simple physical laws that guides the working conditions of a process component. Given the assumption that a fault will lead to plausibility loss, checking the plausibility becomes an important means to reveal the detailed information about faults [8][11].

c. *Signal-based approaches*

The correct sensors are installed in wind turbines instead of input-output models. The sensors measure signals such as electric signals, vibration, and sound signals [57]. It is assumed that information about faults of interest is carried by certain process signals. This approach is used to

get signals which are identified by the corresponding faults [57]. The real time signal and the signals with no fault are compared from the prior knowledge and experiences before a diagnosis decision is made. The schematic diagram of a signal-based approach is presented in Figure 2.03. One example of fault detection using signal-based approaches is artificial neural network [85]. The signal-based fault diagnosis mathematical model is generally divided into time domain method, frequency domain method, statistical method, and time frequency approach [57][65]. This method is widely used in real-time monitoring and diagnosis for induction motors and mechanical components in a system [12]. In the statistical approach, the mean value and the standard deviation are the two basic parameters out of several statistical parameters. These parameters are calculated in the time-domain and are generally used to define average properties of process variables that can cause an occurrence of fault [65]. To perform a quick check on the statistical behaviour of a signal, these parameters can be used. However, in time series analysis, due to its simplicity and ability to show sharp peaks in the frequency domain the autoregressive (AR) is commonly used [65].

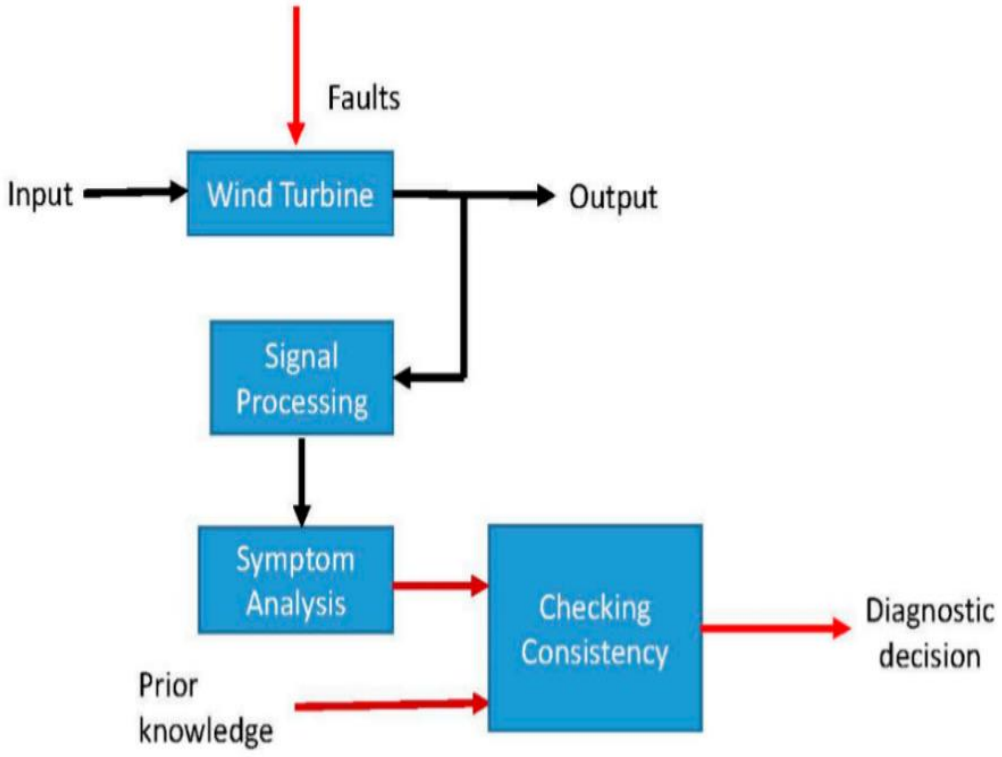

Figure 2.03: Signal based fault diagnosis [57]

A parametric method that is responsible for signal prediction which can be used to predict fault occurrence is known as autoregressive modelling [65]. This approach is preferred in industrial systems where a significant amount of data is used [85].

d. *Model-based approaches*

The hardware redundancy is very expensive and to find a cheaper and better option model-based approach was suggested [40][79][80]. The idea behind this technique is to replace the hardware redundancy using a process model, which is done in the software form on a computer [8][11]. This is suitable for non-stationary wind turbine operations [57][64]. Here the comprehensive model of the system is required to be available. This can be obtained by using system identification techniques. From the model, in the model-based fault detection approach, the wind turbine model and the real time wind turbines will have the same inputs, and the difference in the outputs is monitored [12][57]. The residual is the difference between the system output and the model predicted outputs. The model-based fault detection approach is presented in Figure 2.04. The green part is the subsystem of the wind turbine and the model, and the blue part shows the systems in the green subsystem. 6If the residual is zero, the wind turbine is in a healthy state. Otherwise, if the real time wind turbine outputs are not consistent with the model outputs it signifies there is a fault in the wind turbine system [57]. Not requiring high sampling rate measurements is one advantage of the model-based approach [58][59].

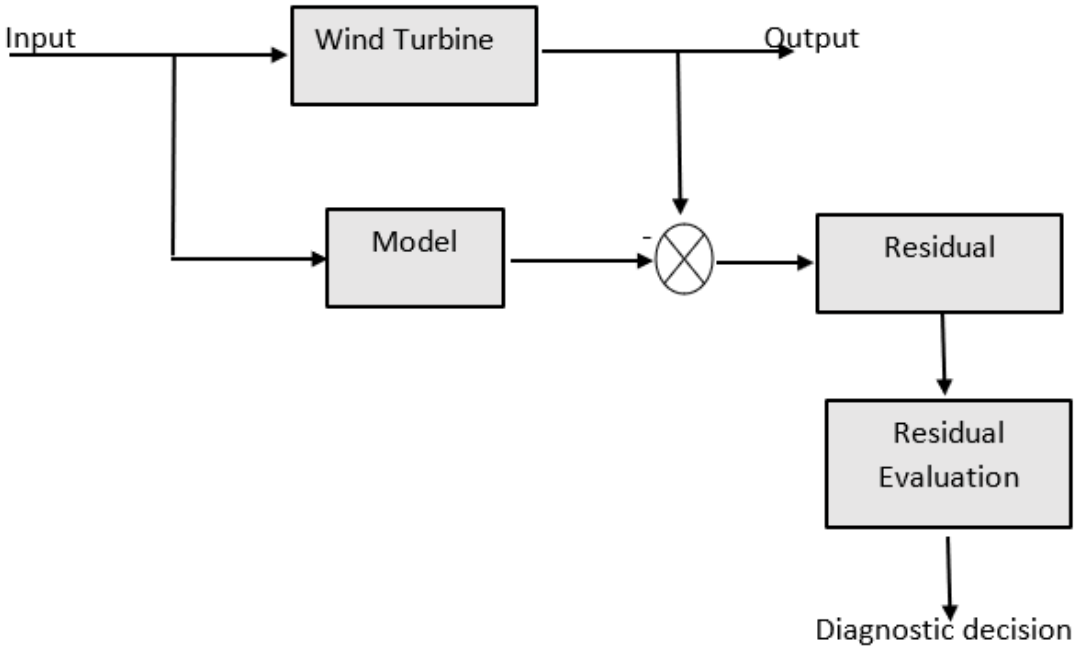

Figure 2.04: Model based fault diagnosis [57]

Observer-based is one of the most used and most effective model-based approaches [62]. An observer is designed to estimate the model output and the residual is observed to see if there is a fault [57].

2.2.2 Fault isolation:

Fault isolation provides the user with information about the fault type. This is a vital step towards fixing online and offline faults [9]. This determines which of the faults has occurred among the possible faults [8]. It determines the location and component in which the fault has occurred [7]. This requires that the fault should be unique [35]. This is implemented based on fault signatures generated by the fault detection module and their relationship with all the considered faults [99]. The fault isolation analysis presented in [9] consists of three parts.

- derivation of adaptive thresholds.
- investigation of conditions for fault insolubility
- fault isolation time computation.

The first part ensures that an incorrect isolation decision will be avoided [9]. This is realized by selecting an adaptive threshold for each fault, such that the residual attached with the isolation

estimator that matches the fault is guaranteed to remain below the threshold [9]. During the adaptive design, there is always a trade-off between false alarms and missed faults [9]. The time interval distance between the fault detection and the determination of its type is known as the fault isolation time [9]. When there is a fault, the fault isolation estimators are activated [9].

2.2.3 Fault identification:

This identifies the fault and indicates the amplitude of the occurred fault [8]. It determines the category of fault and its severity [7][8]. In real application the fault identification phase is rarely used.

2.3 Intelligent fault diagnosis methods

2.3.1 Fault diagnosis using Space vector machine

Support vector machine (SVM) is a special class of machine learning based techniques which are cheap [112]. This statistical approach has become popular due to its preferable effect of classification and regression [112]. This is a family of learning algorithm that have been successfully applied to several problems [111]. SVM are dependent on learning theory and are special for solving learning problems of smaller sample numbers [122]. The learning algorithm of the support vector machine is easy to use and needs few model parameters [111]. Support vector machine are based on the structural risk minimization principle which has improved its generalization property even if the number of samples are reduced [110][109]. It uses structural minimization principles to select functions that have minimal risk bound and the necessary training sample size is smaller [113]. The chances of SVMs to over-fit data is unlikely [113][112]. The Support vector machine-based techniques which have a nonlinear characteristic that tolerates separation of non-linearly separable SVM data have been used extensively to solve classification problems in large feature space [109][110]. SVM are trained as a convex optimization problem which makes them a global solution [113]. The SVM algorithm can improve the detection accuracy which led to it being used for fault detection [109][124]. The support vector machine has a unique principle of classifying data from two different classes [109]. The support vector machine uses temporal correlations for fault detection and attribute correlations for classification for faults [112]. The SVM applied for regression problems are called support vector regression (SVR). This is a special case of SVM that is based on a real value inputs predicts a real value output [113]. SVM algorithm combined with kernel function has the nonlinear attribute and can better handle the case where samples and attributes are massive

[109]. One of the disadvantages of SVM is that it does not perform very well for large data sets [121].

2.3.2 Fault diagnosis using expert system

An expert system is a software system that solves problems that may require significant human expertise using knowledge and inference procedures. it is vital in dealing with problems of incomplete information or large amounts of complex knowledge. It is one of the most vital and active branches of artificial intelligence [123]. The use of expert systems compensates for the scarcity of human experts and can use, preserve, and spread the knowledge and experience of experts [123]. One of the characteristics of the expert system that have made them very useful especially for online operations is that they incorporate symbolic and rule-based knowledge that relates actions and situation. They can explain a line of reasoning. An expert system needs to have a complete understanding of every system component being monitored for it to work effectively. One of the main demerits of expert systems is that human experts don't always think and solve problems by rules and they do not really copy human expert reasoning [123]. In control engineering fault diagnosis is a popular application of the expert system. An expert system has three basic components which are user interface, knowledge base and an inference engine. The user interface changes the user input into a computer language and presents conclusions and explanation to the user. The knowledge base comprises of either shallow knowledge based on heuristics or deep knowledge based on structure or mathematical model. The inference engine supplies inference mechanism to the use of the knowledge directly. This mechanism includes forward and backward chaining, and meta-rules. When a fault is detected, the expert system tries to isolate the cause of the fault using the programmed knowledge base. The knowledge base is a collection of IF/Then rules containing information mapping the system space to the fault space.

2.3.3 Fault diagnosis using Neural network

Neural networks (ANN) for the last few decades have been widely used in modelling and control of nonlinear systems. It is a flexible structure that can make a nonlinear mapping between input and output spaces. The neural network approach is applied on systems whose first mathematical models are too complicated to formulate [120]. The general structure of a NN is stated in Figure 2.05. Neural network application can be classified into three categories. They are outlined below.

I. NN uses its network to differentiate several faults from the normal condition and from one another. This is achieved by using different fault patterns represented in the measured input – output system data either by offline training or by online learning of the fault patterns by an adaptive network. A growing RBF network uses the latter approach to learn various online data patterns. These patterns are represented using the hidden nodes of the network [120]. A faulty condition is believed to occur when a new pattern appears which is significantly different from the existing pattern.

II. NN is used for fault isolation based on the residual generated by a quantitative model-based method.

III. In the third approach, a NN model is used to predict the system output and the residual is generated from the prediction error. For fault isolation, another network is used.

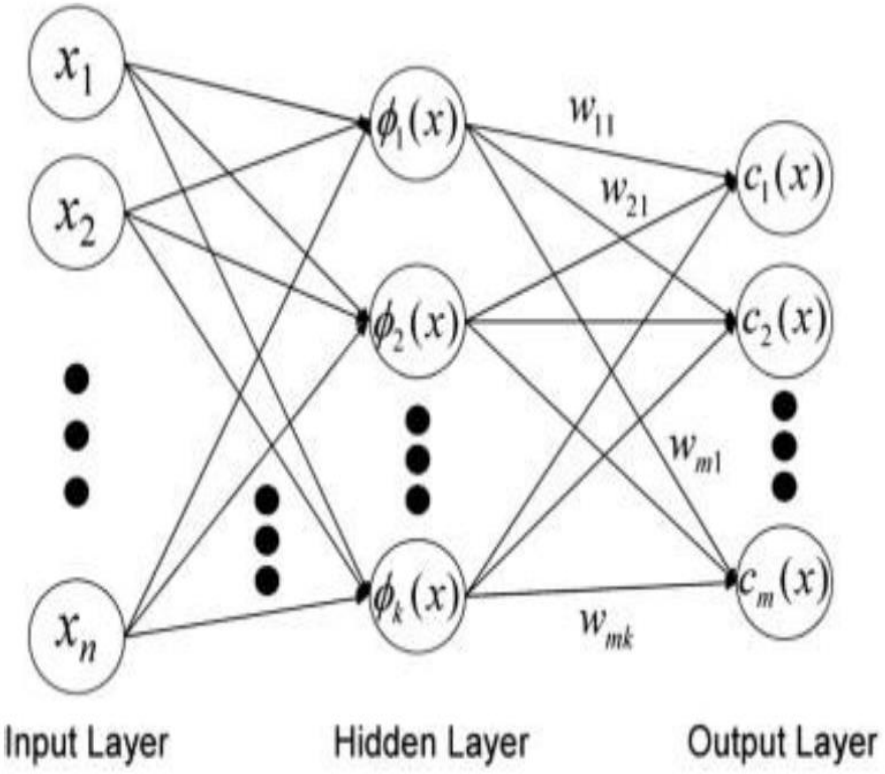

Figure 2.05: General structure of a neural network

The neural network uses the training data to study enough information. The neural network can be changed easily by retraining a new set of training data without modifying the whole program. Neural network can be used to model any nonlinear function which has made it a desirable tool for nonlinear

systems [125]. Neural network does not need the mathematical model of the system to be implemented [125]. Radial basis function (RBF) is one of the types of NN. They have become popular because of the ability of the weights to linearly relate to the objective function and the training is very fast [126]. The modelling of dynamic systems using RBF NN can be carried out using two different modes which are dependent mode and independent mode. In the dependent mode the output of the process is used as part of the network input which makes the model dependent on the process to run [126]. The model here can predict the process output for one step ahead only. The training of the dependent mode is easy for accurate one step ahead prediction. In the independent model, the model output is used as part of the model input instead of the process output. The model can predict for long steps provided the input is available. The structures of the dependent and independent mode are illustrated in Figures 2.06 and 2.07.

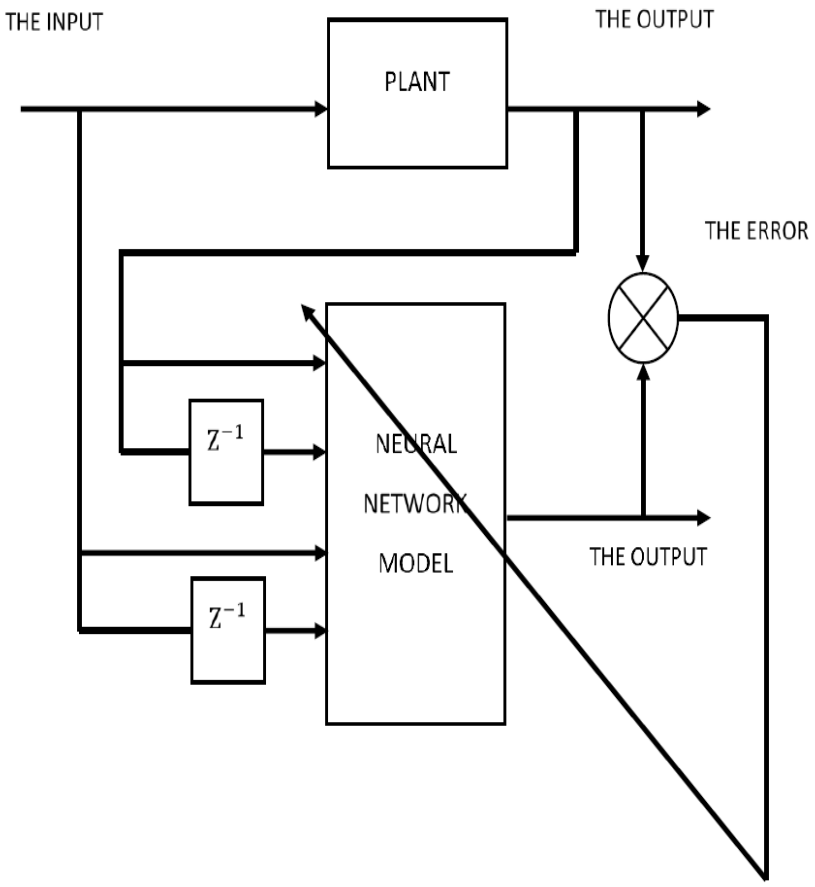

Figure 2.06: Dependent model structure

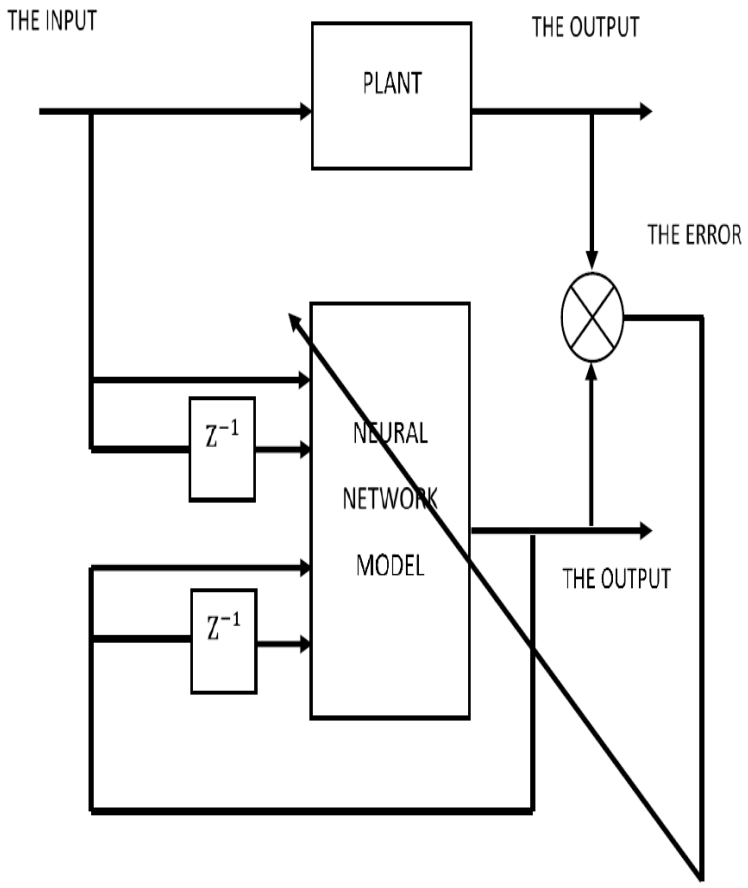

Figure 2.07: Independent model structure

2.4 Fault tolerant control (FTC)

The popularity of the FTC in recent times has been due to the need to increase the reliability and possibility of a distributed energy system [39][81][84]. Also, the reason for the popularity is that faults exist in the real-life world and may lead to loss of properties [68][69]. For an improved efficiency of the system, the reliability of the system can be improved by fault tolerant control (FTC) [7][87]. A fault tolerant control system should have the ability to detect faults early without human intervention and accommodate the effects of the faults on the systems in such a way that the system can maintain working while the performance provided is acceptable [5][7][10]. Despite the presence of faults an FTC is an effective approach to deal with these faults and guarantee stability, tracking performance [68][71][72]. Different fault tolerant control approaches have been established in the literature to improve the safety and reliability, and to keep the system's stability and convergence

under some conditions. From a control strategy point of view, fault tolerant control can be divided into two categories, passive and active methods [13].

2.4.1 Passive FTC method:

Here both the normal and faulty operation system performances are dealt with by a single controller with fixed structure without any feedback information from the fault [5][13][101]. The fault information is assumed to be on-line [37]. The FTC system here does not need the faulty information to control the system and is related closely to robust control [74][75]. A robust control is a fixed controller that is robust against any possible fault in a system [73][90][104]. The passive FTC uses the concept of hardware redundancy. The passive FTC is popular due to its simplicity in implementation, low computation load and rapid response to faults [77][78]. Its limitation is that the design needs to have prior knowledge of possible faults occurrence [37]. Another limitation is that if a failure that is not anticipated during the design occurs, the performance of the closed-loop system may not be guaranteed, and its stability will be affected. The dependence on hardware redundancy is another limitation of this method [73][83]. When using hardware redundancy, the product cost, product weight and product size will increase. These limitations found in the passive method motivated researchers to develop some other method of fault tolerant control [13]. Sliding mode control is one of the examples of the passive FTC [67]. This second method known as the active FTC is discussed below. Figure 2.08 shows the structure of the passive FTC method.

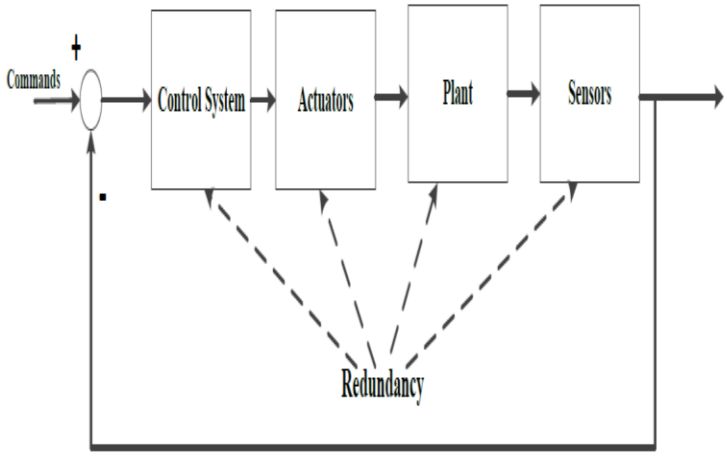

Figure 2.08: Structure of the passive FTC [73]

2.4.2 Active FTC method:

Here this method is confident that the controller can change the structure to actively acquire information suitable on faults and the system state can change to perform prompt dynamic correction to guarantee that the system stability is maintained after a fault occurs [90][70]. In developing the control signal here, the following phases are considered; In the first phase, fault detection and isolation scheme are designed to make available the fault information. False alarm or inaccurate information will have a direct effect on the performance of the FTC [73]. The fault information obtained in phase one is used as a feedback in phase two to adjust the control input and the controller structure. The active FTC will be robust against imperfect fault information. One limitation of the active fault tolerant control is that the performance is degraded due to time delay caused by the online fault diagnosis and controller reconfiguration. The system becomes unstable or out of control if the delay of the fault diagnosis is too long. However, if the fault detection and isolation process avoid making an incorrect or delayed decision, active fault tolerant techniques can re-configure the control system structure and controllers based on fault information and lead to better performance of the closed-loop system [13]. System performance in active fault tolerant control depends on how accurate the fault information obtained is. With accurate information, the performance becomes more active than the passive fault tolerant system, which makes it favourite in practice [14]. The fault detection and fault isolation produce the information used for controller redesign as shown in Fig 2.09

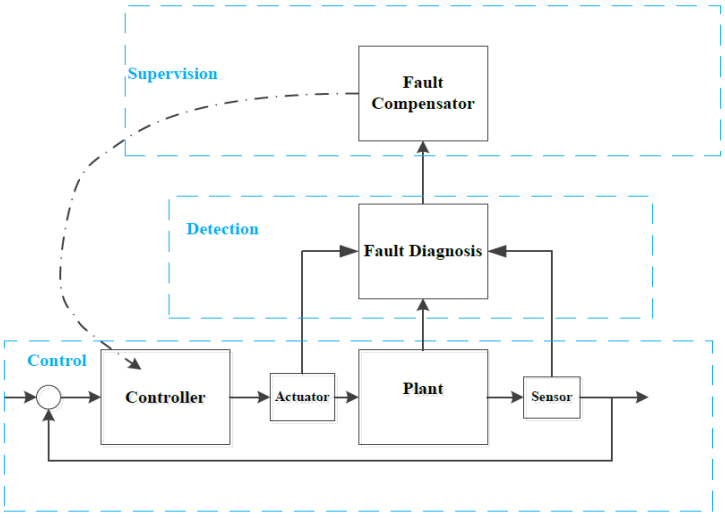

Figure 2.09: Structure of Active fault tolerant controls [73]

2.5 Sliding mode control

Sliding mode control (SMC) is a type of variable structure control (VSC) method used to control high-order nonlinear systems under certain conditions [15][105]. It is nonlinear robust control method targeted at removing the effects of uncertainties or external disturbances that are bounded by means of discontinuous controller [16][17]. It is characterised by a decision rule and feedback control laws [16][103][17]. This technique is expected to drive the system state variables to equilibrium using a discontinuous feedback control law [40][91]. Sliding mode control consists of the following steps:

- Designing a suitable sliding surface that has good performance and convergence properties when the dynamics are reduced and are also independent of the disturbances [17].
- Design of a discontinuous term in the control law with the task that in finite time, the trajectories of the system are pushed to the sliding surface and remain on it [17][92].

Sliding mode control is a high frequency switching control that is not affected by internal or external structured disturbances [18][28]. The choice of the switching function directly tailors the dynamic behaviour of the system. Due to the advantages, which include fast response, easy implementation and robustness to external disturbances, interest in sliding mode control method has increased significantly when compared with other control methods [15][16]. Generally, a system under sliding mode control design exhibits asymptotic stable performance [6][15]. Due to the high frequency switching of the sliding mode control signal it has a chattering effect as a disadvantage [19][95]. Sliding mode controller does not assure that the sliding surface offers a finite time convergence [14][29]. Sliding mode control is divided into linear and terminal sliding mode [41][34]. In the linear sliding mode, the sliding surface are linear functions that contain the state of the system which are suitable with low requirements on state accuracy [41][93]. The sliding surface used in linear sliding mode is the hyperplane [42][43][44]. The linear sliding mode cannot converge in finite times when used in complex nonlinear systems even though it ensures asymptotic convergence of system state to the equilibrium point [41][42]. The design task for a linear sliding mode is design of control strategy and sliding mode surface [45].

2.5.1 Terminal sliding mode:

To solve the disadvantage experience with the conventional SMC an advanced SMC known as terminal SMC (TSMC) has been developed recently as an active method [14][49][50]. Instead of using linear surface, TSMC uses nonlinear sliding surface to enable the system to guarantee finite

time convergence [14][93][15]. In recent years terminal sliding mode control has attracted much interest due to its advantages in making nonlinear systems efficiently achieve a fast and finite-time convergence and its ability to be robust to system uncertainties [1][52]. Two main problems have often been associated with terminal sliding mode which are singularity and chattering. The existence of negative fractional powers in the TSMC causes the singularity. This means that in the state space, the terminal sliding mode control may need to be infinitely large in order maintain the ideal terminal sliding mode motion [45][82]. A couple of researchers have conducted research to remove this singularity problem which has limited the application of terminal sliding mode. The method proposed in [46] was able to avoid the singularity problem completely but the method is suitable only for the second order system and some class of high order systems. A derivative and integral terminal sliding mode control was developed in [47] but they are also for a special class of high order system and second order system. In [48] the singularity problem was further investigated, and the result showed that if the terminal sliding mode manifold is chosen as reduced order the singularity areas will always exist in the state space.

Another problem to worry about is the chattering problem. The switching function adopted by the terminal sliding mode control can cause high frequency oscillation in the system state [34][96]. This high oscillation is what is known as chattering. This can be simply referred to as the result caused by the existence of discontinuous terms in TSMC. Chattering can lead to mechanical wear of the actuators. In [34][46][51], the state travels away from the sliding surface to achieve finite-time convergence to a bounded layer around zero and not the origin so that its characteristics are lost. This is because a reduced TSMC was used.

2.5.2 Full order terminal sliding mode control:

A novel concept of full order terminal sliding mode control which overcame the two problems of chattering and singularity was proposed in [34]. This method is targeted at achieving a finite time convergence to the origin. In this concept, the order of the designed FOTSM is the same as that of the system model. The full order terminal sliding mode proposed in [34] has exciting features such as avoidance of singularity and chattering. The singularity problem was avoided by preventing differentiating terms with fractional power in sliding mode manifold from deriving the control laws and by applying a continuous control strategy, the chattering in both linear system and terminal sliding mode is resolved [34]. The system behaves as a desirable full-order dynamic rather than a desirable reduced order dynamic with the control strategy in place. One major demerit of the method proposed in [34] is that it requires prior knowledge of the system model and the bounds of the system

uncertainties derivatives. This may be difficult to obtain in real life applications. In [94] a full order terminal sliding mode control is proposed for a class of nonlinear system. The systems behaved as a desirable full-order dynamic rather than reduced-order dynamics during the ideal sliding motion [94]. The control strategies do not include the fractional power terms and the control signals are continuous [94]. This guarantees singularity is avoided, and chattering is attenuated [94]. The finite-time convergence of the system is guaranteed with the full-order terminal sliding mode control even in the presence of error [94].

2.6 FD and FTC for wind turbine systems

Wind turbine systems have attracted more attention from energy suppliers around the world; this is due to increased government investments and promotion to go green [53]. Megawatt-sized wind turbines are expensive; therefore, it is anticipated that their energy production within the short downtimes will be worth the investment [3]. This is hard to achieve without the introduction of fault detection, isolation and accommodation systems into the turbines [3][54]. Most of these operating conditions increase the possibility of the system failures characterized by unpredictable changes in the system dynamics [99]. The wind speed is divided into four regions according to the control objective and wind turbine operation [56]. This is outlined in Figure 2.10 where the cut in and cut out wind speeds are denoted as min v and max v. In region 1, the turbine is not at work and the wind speed is lower than the cut in value [98]. In Region II, the wind speed is below the rated speed [56]. The control objective in this region is to capture all possible wind power attaining global maximum power with the pitch angle set at zero. Here the generator load torque is regulated [25][55]. In Region III, the wind speed is higher than the rated speed. This is between nominal speeds and cut-out. The wind turbine power output is maintained at the rated power [21]. This is to ensure that the wind turbine works within its limit. Region IV denotes the high wind speed which leads to shutdown of the turbine to avoid damages [25][54]. The wind turbine operations modes are presented in Figure 2.10.

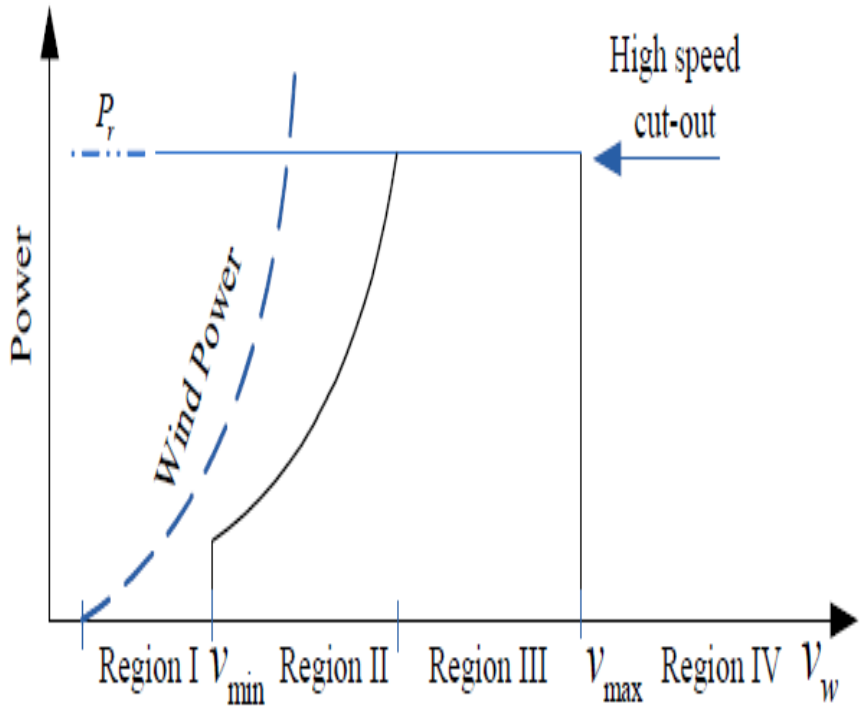

Figure 2.10: Wind turbine operations modes [25]

Table 2.0 outlines some of the faults experienced in wind turbine systems and the main causes of the faults are presented in Table 2.1.

Table 2.0: Wind turbine system faults and percentage [57][85]

s/n	Faults	Percentage (%)
1	Electrical system	23
2	Gear Box	4
3	Mechanical Break	6
4	Sensors	10
5	Hydraulic system	9
6	Yaw system	8

7	Drive train	2
8	Rotor Hub	5
9	Structural parts/housing	4
10	Generator	4
11	Plant control system	18
12	Rotor blade	7

Table 2.1: Typical fault causes in a wind turbine [57]

s/n	Fault type	Causes of faults
1	Blades and rotors faults	Corrosion of blades and hub; reduced stiffness; increased surface roughness; deformation of the blades; crack errors of pitch angle; and imbalance of rotors, etc
2	Gear box faults	Shaft imbalance and misalignment, high oil temperature, poor lubrication and broken shaft.
3	Generator faults	Generator overheating, abnormal noises. Generator vibrating excessively and insulation damage.
4	Bearing faults	Unplanned stress could cause overheating and premature wear.
5	Faults on main shaft	Corrosion, crack and coupling failure
6	Hydraulic faults	Sliding valve blockage

7	Mechanical braking system faults	Failures of the Hydraulic and exceeding the wind speed limit.
8	Faults on tower	Improper installation and loading, harsh environment.
9	Electrical system faults	Electric leaks and cold-solder joints
10	Sensors faults	Malfunction of physical failure of a sensor.

The adoption of fault-tolerant controller designs with timely fault detection, isolation and successful controller reconfiguration is very important in wind turbine systems [8]. The ability of a control system to continue operation despite the presence of a fault is known as fault tolerant control. The fault tolerant system provides the ability of a system to detect a fault early without human intervention and accommodates the effects of the faults on the system in such a way that the system can maintain working while the performance provided is acceptable [5]. By Interacting with any pre-existing controllers to cancel the effects of faults, the fault tolerant control compensates for the faults in the wind turbine [61]. In [88] an unknown input observer (UIO) was used to estimate faults in the converter and accurately isolate if it is an actuator or sensor fault. The choice of the UIO was made because the wind speed is assumed unknown, and its measurement is affected by turbulence around the rotor plane [88]. In [97] an unknown input observer is designed to detect in a specific benchmark model three sensor faults for fault detection and accommodation of wind turbines. The proposed scheme within the simulation time was able to detect sensor faults in question but the converter fault was detected outside the simulation time [97]. In [9], a robust detection and isolation scheme was developed for abrupt and incipient faults in nonlinear Systems. A detection and approximation estimator are used for online health monitoring. In [60] sensor and actuator faults diagnosis for a wind turbine system was achieved using robust observer and filter. The study considered individual pitch control as the load reduction. The observer was designed to be sensitive to faults but insensitive to disturbances. In [23] a group of unknown input observers were used for fault diagnosis in the generator speed sensors and the rotor of the fault detection isolation benchmark model. Some methods use analytical redundancy of the system for fault estimation and the design of the virtual actuators or sensors [24]. In [129], a wind turbine model was designed

based on the closed loop identification technique, where the wind dynamics were included. Dual Kalman filters was used to generate residuals for the fault detection. Both faults considered here are additive and multiplicative. [130] talked about the Fault Detection and isolation of the wind turbine benchmark using an Estimation-based Approach. A fault detection estimator was used to monitor the occurrence of a fault, and a bank of fault isolation estimators were employed to determine the exact fault type.

2.7 Summary

In dynamic systems, a fault can be defined as an unpermitted deviation of the structure of a system from the nominal situation. This fault diagnosis can be achieved by the combination of fault generation and residual evaluation. Fault diagnosis is made up of fault detection, fault isolation and fault accommodation. The objective of fault detection is to determine the occurrence of a fault. Fault isolation provides the user with information about the fault type. Fault tolerant control can be divided into two categories, passive, and active methods. Sliding mode control (SMC) is a type of variable structure control (VSC) method used to control high-order nonlinear systems under certain conditions. A novel concept of full order terminal sliding mode control which overcame the two problems of chattering and singularity was proposed.

CHAPTER 3
Wind Turbine Modelling

3.1 Introduction

In the wind turbine system proposed, a benchmark model of the wind turbine with physical parameters as used in [3] will be used. A linear mathematical model for the wind turbine pitch system and drive train will be proposed. The wind turbine is designed in a way to conveniently allow electrical energy to be generated from the wind's kinetic energy. A MATLAB and Simulink model will be developed using the benchmark mathematical model in which an actuator fault and sensor faults will be introduced. The following sub-systems make up the wind turbine system; Aerodynamics, Pitch Actuator, Drive-train System, the generator and converters. A block diagram of this wind turbine is shown in Fig.3.01.

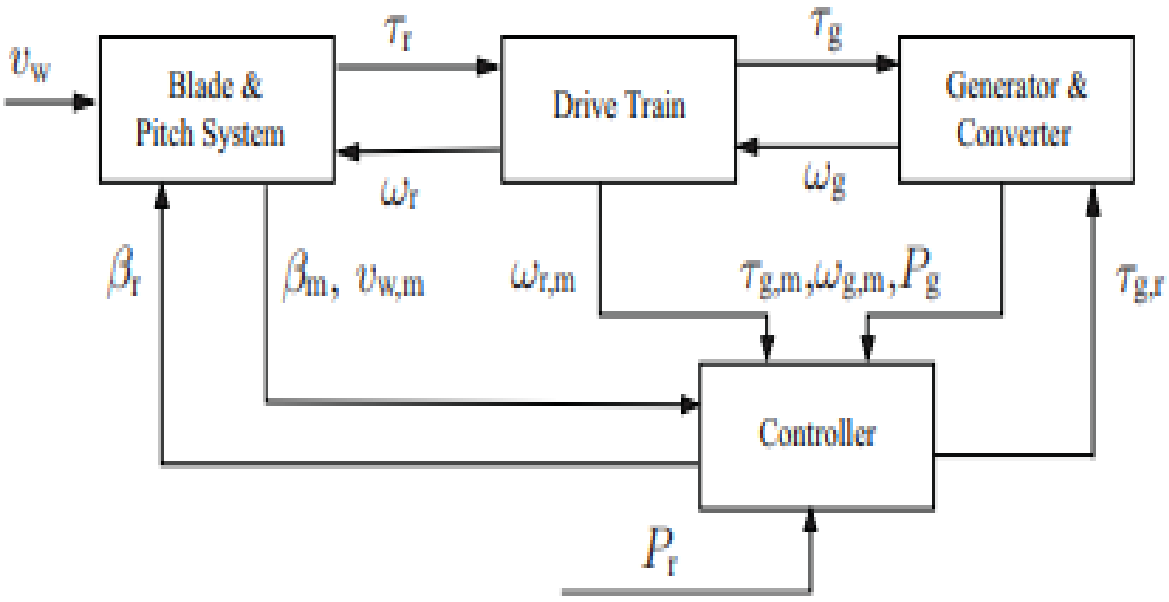

Figure 3.01: Block diagram of the wind turbine benchmark [5]

3.2 Pitch and blade model

The aerodynamic model and the pitch model make up the pitch and blade model. The aerodynamic model which is used to determine τ_r needs wind speed and pitch models [3].

3.2.1 Aerodynamic model

The drive train is used to transfer aerodynamic torque from the rotor to the generator [3]. The aerodynamic torque is given by (3.01) [1],[3].

$$\tau_r(t) = \frac{\rho\pi R^3 C_q(\lambda(t),\beta(t))v_w^2(t)}{2} \quad (3.01)$$

here $C_q = \frac{C_p}{\lambda}$, C_p is the power coefficient. This is the quantity of wind turbine efficiency used by the wind turbine. ρ is the air density, R is the radius of the blade, C_q is the torque coefficient, β is the pitch angle and λ is the tip speed ratio. The ratio of the translational speed at the wind turbine blade to the actual velocity of the wind is called the speed ratio.

$$\lambda = \frac{w_r R}{v_w}$$

Power in the wind is given by the rate of change in energy [3].

$$P_w = \frac{dE}{dt} = \frac{1}{2}\rho A v_w^3 \quad (3.02)$$

Where $A = \pi R^2$, R is the radius of the blade

According to Betz limit, a wind turbine can extract no more than 59.3% of the energy carried by the wind [3]. The C_p value is unique to each turbine type as no wind turbine can operate at the C_{pmax} value [3]. C_p is dependent on the tip-speed ratio λ and blade pitch angle β. This is expressed in (3.03) [1], [3]. Due to factors like gear box, bearings, and generator, only 10-30% of the power of the wind is eventually converted into useable electricity. Hence the available power from the wind is given by (3.04) [3].

$$\begin{cases} C_p(\lambda,\beta) = 0.22\left(\frac{116}{m} - 0.4\beta - 5\right)e^{-12.5/m} \\ \frac{1}{m} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \end{cases} \quad (3.03)$$

w_r =Mechanical angular velocity in rad/sec. v_w =Wind speed in m/s

$$P_{avail} = \frac{1}{2}\rho A v_w^3 C_p \quad (3.04)$$

3.2.2 Pitch system

The pitch system is a hydraulic system which is modelled as a second order system between the measured pitch angle β and the reference signal β_r [1][3]. Figure 3.02 displays the block diagram of the pitch system.

$$\ddot{\beta}(t) = -2\dot{\beta}(t)\zeta\omega_n - \beta(t){\omega_n}^2 + {\beta_{\mathrm{r}}(\mathrm{t})\omega_n}^2 \quad (3.05)$$

Where β_{r} is the desired pitch angle and β is the actual pitch angle ω_n and ζ are the natural frequency and damping ratio of the pitch actuator respectively [3].

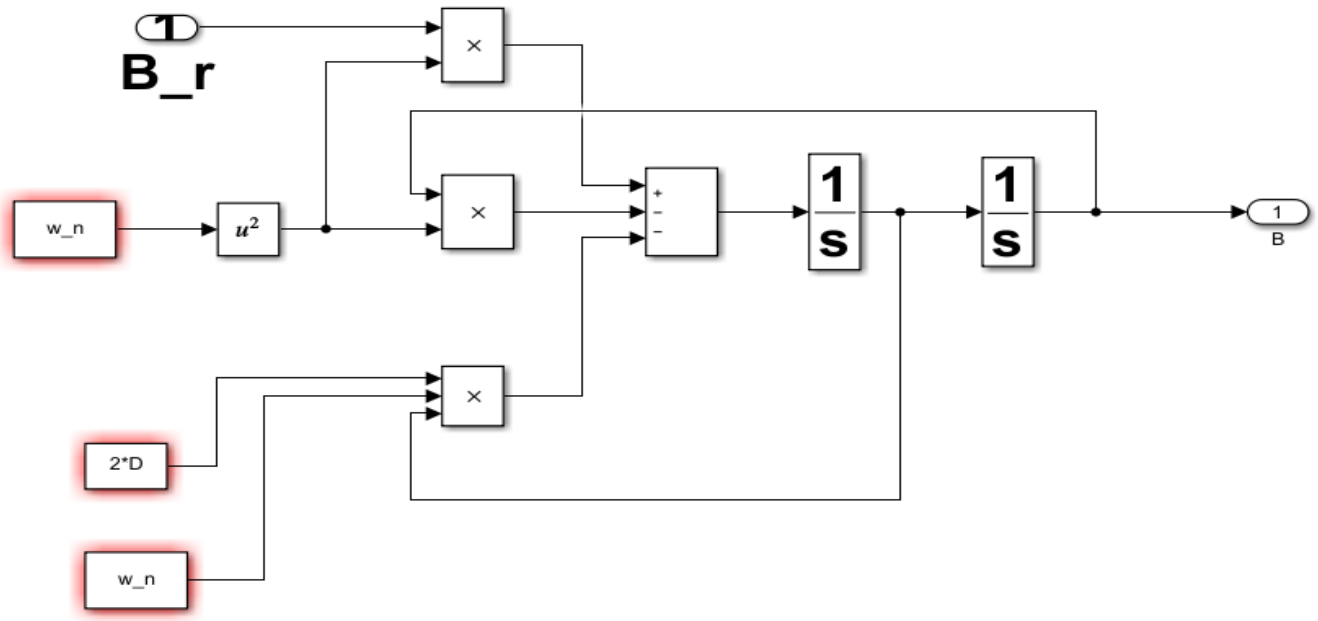

Figure 3.02: Block diagram of the pitch system model

3.3 Drive Train

The drive train model is modelled as a rotational 2-mass, 1-spring.1-damper system. The drive train of a wind turbine system consists of a main gear box, low speed shaft (turbine) and a high-speed shaft (generator) whose inertias are denoted as J_r and J_g respectively [1][26]. Its aim is to transfer torque from the low-speed shaft (turbine) to the generator using a gearbox [1]. The drive train model is presented in Figure 3.03

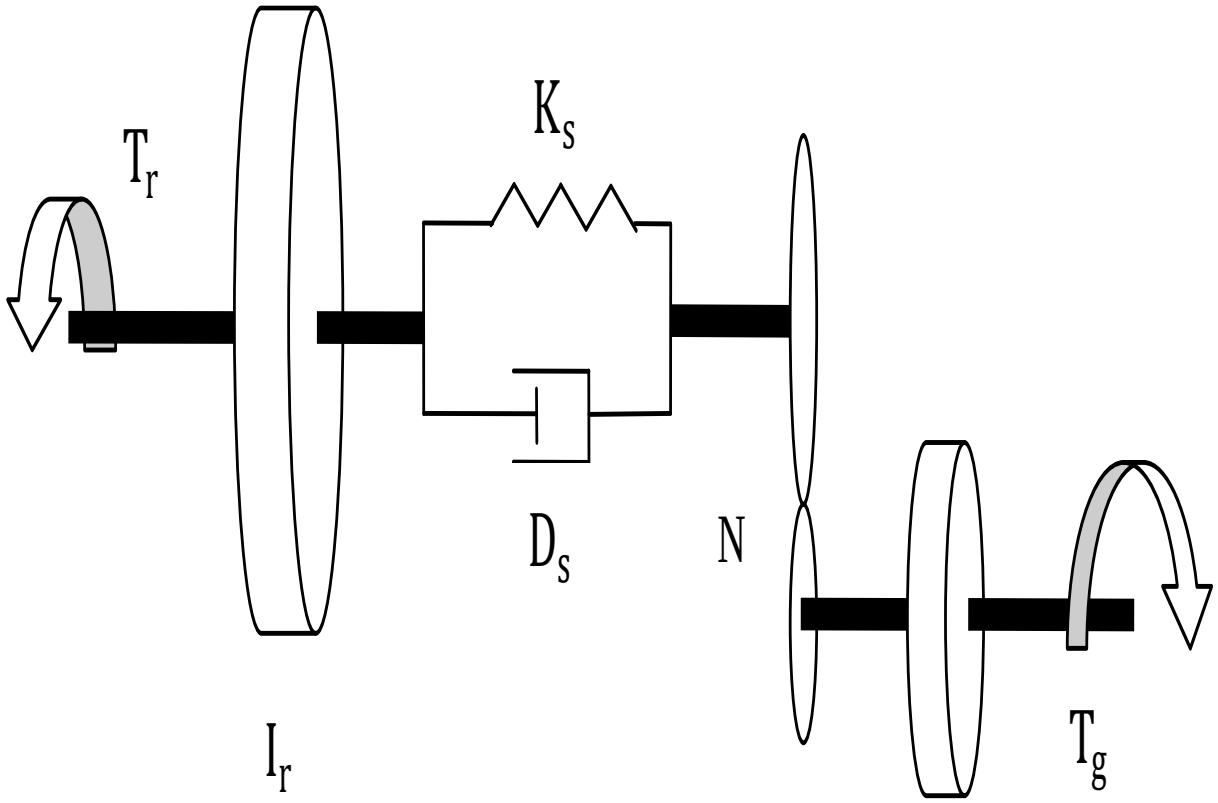

Figure 3.03: Drive train dynamics [1]

In Figure 3.4, the shaft on the rotor side is assumed to be flexible and the shaft on the generator side is rigid. The gear unit is assumed to be loss-free. N is the gear ratio between the rotor and the generator. The drive train systems differential takes the form of (3.06) to (3.07) [1]

$$\dot{\omega}_r = -\frac{D_s}{J_r}\omega_r + \frac{D_s}{J_r N}\omega_g + \frac{K_s}{J_r}\theta + \frac{\tau_r}{J_r} \tag{3.06}$$

$$\dot{\omega}_g = -\frac{D_s}{J_g N}\omega_r - \frac{D_s}{J_g N^2}\omega_g - \frac{K_s}{J_g N}\theta - \frac{\tau_g}{J_g} \tag{3.07}$$

3.4 Generator and Convertor Models

The generator and converter dynamics are modelled by a first-order transfer function. The kinetic energy of a mass in motion is given by $\frac{1}{2}mv_w^2$.

Where m is the mass. v_w is the wind speed. Power in the wind is given by the rate of change in energy

$$P_w = \frac{dE}{dt} = \frac{1}{2} v_w^2 \frac{dM}{dt} \quad (3.08)$$

Mass flow rate`= $\frac{dM}{dt} = \rho A \frac{dx}{dt}$

Where $\frac{dx}{dt} = v_w$

Therefore $\frac{dM}{dt} = \rho A v_w \quad (3.09)$

Substitute (3.09) in (3.08)

$$P_w = \frac{1}{2} \rho A v_w^3 \quad (3.10)$$

$$P_{avail} = \frac{1}{2} \rho A v_w^3 C_p \quad (3.11)$$

Where

$$A = \pi R^2$$

R is the radius of the blade

C_p is the power coefficient and is a function of speed ratio λ and pitch angle β

$$\lambda = \frac{v_w}{w_r R}$$

w_r =mechanical angular velocity in rad/sec

v_w =wind speed in m/s

$$w_r = \frac{2\pi n}{60}$$

Where n= rotational speed

(3.11) can be rewritten as

$$P_{avail} = \frac{1}{2}\rho A v_w^3 C_p(\lambda, \beta)$$

Where $C_p(\lambda, \beta)$ can be determined using (3.03)

The power converter dynamics is modelled by a first order transfer function.

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

Where a_{gc} is the inverse of the first order time constant and $\tau_{g,r}$ is the reference torque to the generator [3].

$$\tau_g s + \tau_g a_{gc} = \tau_{gr} a_{gc} \tag{3.12}$$

Using inverse Laplace transform (3.12) becomes

$$\dot{\tau}_g(t) + \tau_g a_{gc} = \tau_{gr} a_{gc}$$

$$\dot{\tau}_g(t) = -a_{gc}\tau_g(t) + a_{gc}\tau_{g,r}(t) \tag{3.13}$$

The power produced by the generator is given by:

$$P_g = \eta_g \omega_g(t) \tau_g(t) \tag{3.14}$$

Where η_g is the efficiency of the generator [1].

The benchmark model is presented in Figure 3.04 with the wind speed represented by a random number signal. The other components as explained in sections 3.2, 3.3 and 3.4 are combined to achieve this non linear benchmark model of the wind turbine.

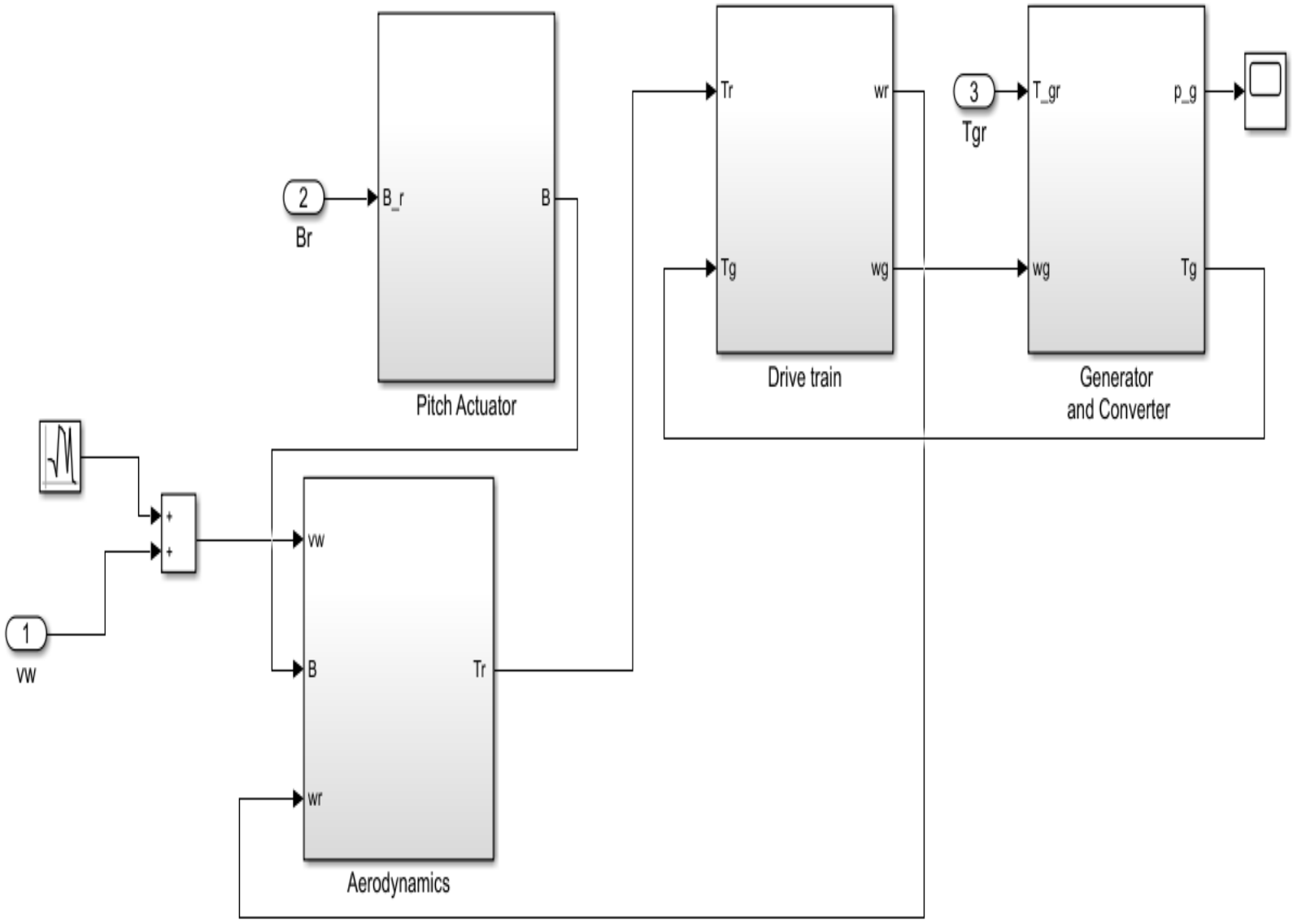

Figure 3.04: Nonlinear Simulink model of the wind turbine

The pitch angle, rotor speed and generator speed are displayed in figure 3.05, figure 3.06 and figure 3.07 respectively.

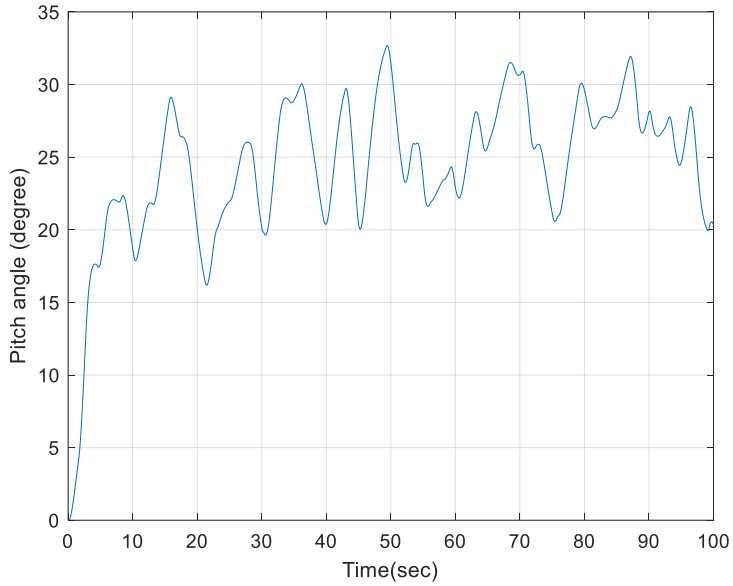

Figure 3.05: Pitch angle of the wind turbine benchmark model

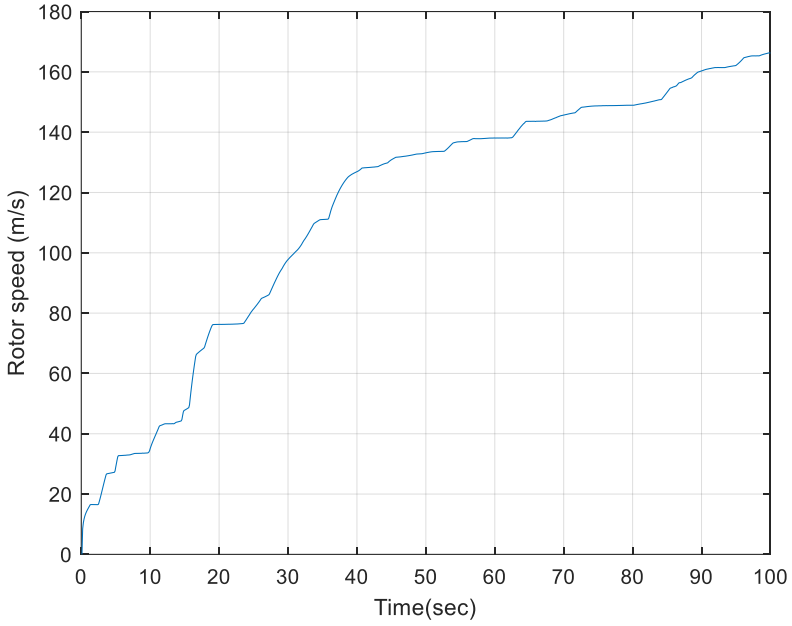

Figure 3.06: Rotor speed of the wind turbine benchmark model

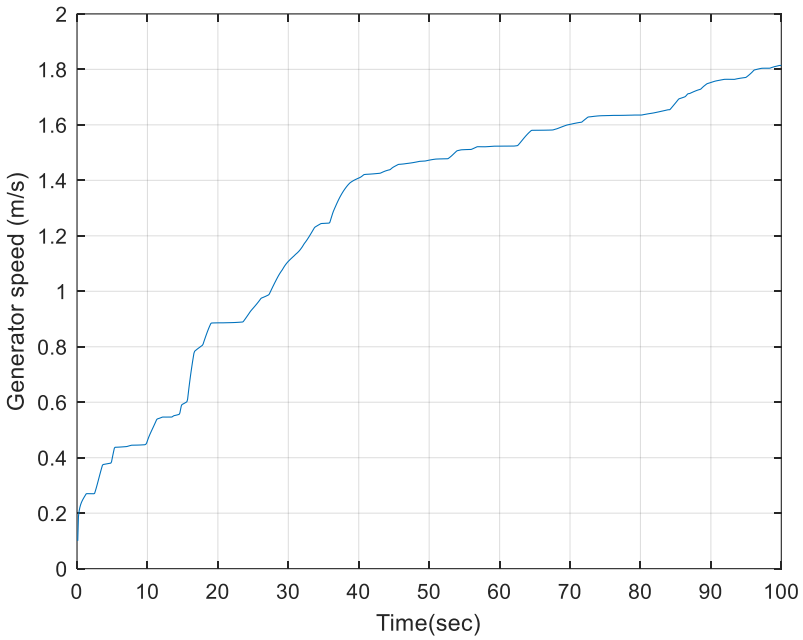

Figure 3.07: Generator speed of the wind turbine benchmark model

3.5 Combination of Pitch blade and drive train dynamics

The linear model of the wind turbine benchmark model is derived by combining the pitch dynamics and the drive train dynamics in (3.15) to (3.19). These are combined to set up the state space equation in (3.20) and (3.21).

$$\dot{\beta} = \dot{\beta} \tag{3.15}$$

$$\dot{\omega_r} = -\frac{D_s}{J_r}\omega_r + \frac{D_s}{J_r N}\omega_g - \frac{K_s}{J_r}\theta + \frac{1}{J_r}T_r \tag{3.16}$$

$$\dot{\omega_g} = \frac{D_s}{J_g N}\omega_r - \frac{D_s}{J_g N^2}\omega_g + \frac{K_s}{J_g N}\theta - \frac{1}{J_g}T_g \tag{3.17}$$

$$\dot{\theta} = \omega_r - \frac{\omega_g}{N} \tag{3.18}$$

$$\ddot{\beta} = -{\omega_n}^2\beta - 2\zeta\omega_n\dot{\beta} + {\omega_n}^2\beta_r \tag{3.19}$$

The inputs are the rotor torque, pitch angle reference and the generator torque. The outputs are the rotor speed, generator speed and the pitch angle.

$$\dot{x} = Ax + Bu + Ed(t) + Gf_a \tag{3.20}$$

$$y = Cx + Qf_s \tag{3.21}$$

Where $x^T = \begin{bmatrix}\beta & \omega_r & \omega_g & \theta & \dot{\beta}\end{bmatrix}$, $u^T = \begin{bmatrix}T_r & T_g & \beta_r\end{bmatrix}$, $y^T = [\omega_r \quad \omega_g \quad \beta]$.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & -\frac{B_{dt}}{j_r} & \frac{B_{dt}}{j_r N} & -\frac{K_s}{j_r} & 0 \\ 0 & \frac{B_{dt}}{j_g N} & -\frac{B_{dt}}{j_g N^2} & \frac{K_s}{j_g N} & 0 \\ 0 & 1 & -\frac{1}{N} & 0 & 0 \\ -{\omega_n}^2 & 0 & 0 & 0 & -2\zeta\omega_n \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{j_r} & 0 & 0 \\ 0 & -\frac{1}{j_g} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & {\omega_n}^2 \end{bmatrix} \mathrm{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$x, u \; and \; y$ are the system state, input and output signals. A=5 × 5 , B=5 × 3 , C=3 × 5. $d, f_s and \; f_a$ represents the disturbances, sensor faults and actuator faults. A, B, C, E and G are all known matrices.

3.6 Electro-hydraulic servo pitch system for wind turbine

The variable-speed wind turbines have experienced fast growth and have been preferred in big wind power applications [127]. This could be attributed to longer life span, improved power quality, high power efficiency and cost efficiency when compared to the fixed speed wind turbine. One other advantage of variable speed turbines when looked at in a big range is their ability to cope with the frequent change in the wind speed. The variable speed wind turbine pitch system permits the turbine to operate in the partial load region with efficiency. There are two categories of the pitch operating systems in general. They are electro-mechanical and hydraulic pitch system. The electro-mechanical system engages an electrical servomotor normally to drive the pitch turbine blades. The hydraulic pitch uses a servo valve-controlled hydraulic cylinder to convert through a slider clank mechanism the linear displacement of the hydraulic cylinder into the turbine blades pitch angle. Electromechanical pitch system suffers a couple of disadvantages like limited torque and low power to mass ratio. The hydraulic pitch has a vital advantage like easy installation, high reliability, self-cooling, and large robustness while it suffers low accuracy in the pitch angle. The low accuracy is caused by the nonlinear transformation between hydraulic piston displacement and pitch angle. The

pitch angle system proposed in this research is an Ac servo motor driven hydraulic pump which is used to drive a hydraulic motor. It consists of a hydraulic motor, pitch gear set two relief valves and a variable speed hydraulic pump. The hydraulic motor is driven by the hydraulic pump. The relief valves are linked across the hydraulic lines to create an upper limit to the pipeline pressure. For the turbine blades the high speed and low torque of the hydraulic motor is converted into the low speed and high torque through the pitch gear set. The schematic diagram of the hydraulic pitch drive system can be found in Figure 3.08.

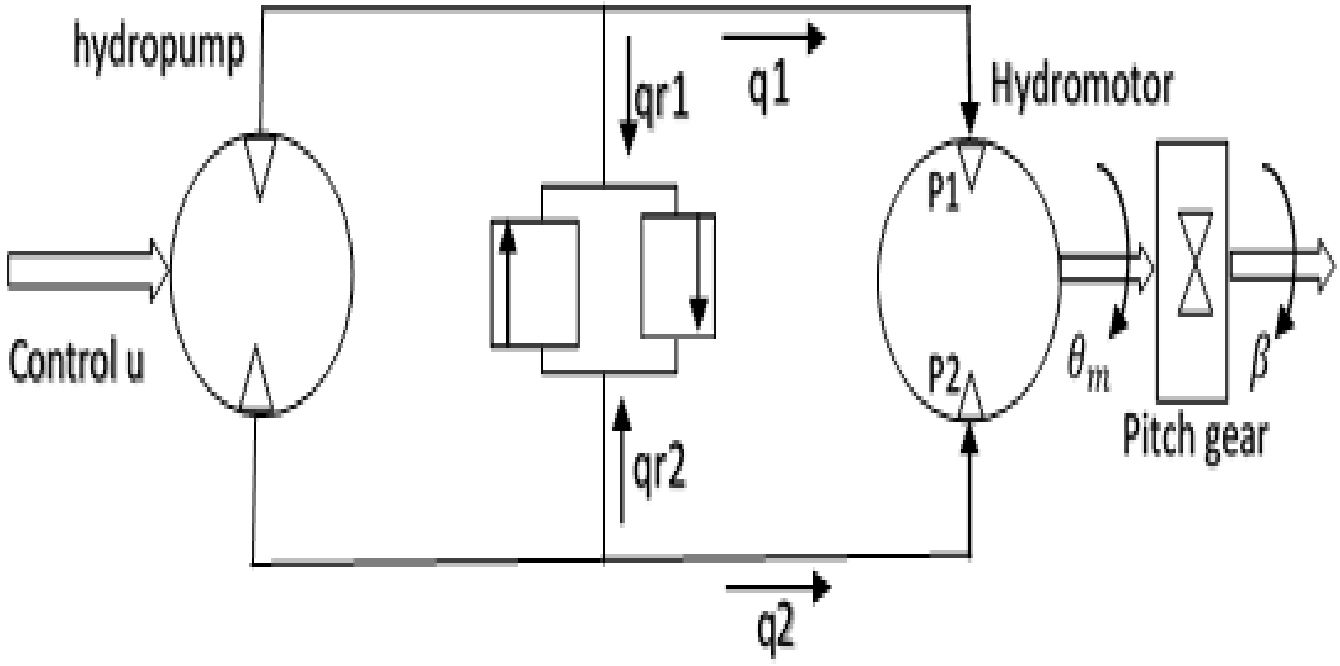

Figure 3.08: Schematic of pitch angle hydraulic drive system

A closer look at Figure 3.08 shows the manipulated variable is the voltage signal of the motor input while the controlled variable is the pitch angle of the blade.

3.6.1 Dynamic modelling

The input control voltage to the AC servo motor is transferred into the rotational speed of the hydraulic motor. Since the servo motor dynamics have much faster response than other components in the pitch system. For simplicity purposes the motor dynamics are neglected and only the static gains remain

$$w_p(t) = K_a u(t) \tag{3.22}$$

Where w_p is the rotational speed of the servo motor.

$K_a$is the gain of the servo control unit and the input control voltage is u.

The flow in the inlet and outlet pipe of the hydraulic pump is represented by (3.23) and (3.24);

$$q_1(t) = D_p w_p(t) - C_{tp}[P_1(t) - P_2(t)] - q_{r1}(t) \quad (3.23)$$

$$q_1(t) = D_p w_p(t) - C_{tp}[P_1(t) - P_2(t)] - q_{r2}(t) \quad (3.24)$$

Where

q_1, q_2 is the hydraulic flow rates.

D_p is the volumetric displacement of the pump.

P_1, P_2 is the hydraulic motor chamber pressures.

C_{tp} is the leakage coefficient of the hydraulic pump.

q_{r1}, q_{r2} is the flow rates through the two relief valves.

The chamber pressures of the hydraulic motor under normal operating conditions are lower than the preset pressures of the relief valves. This has led to the flow rates q_{r1}, q_{r2} being set as zero. The hydraulic motor chambers continuity equations are written below;

$$q_1 = D_m w_m(t) + C_{tm}[P_1(t) - P_2(t)] + \frac{v_1 + D_m\theta_m(t)}{\beta_e}\dot{P}_1(t) \quad (3.25)$$

$$q_2 + \frac{v_1 - D_m\theta_m(t)}{\beta_e}\dot{P}_2(t) = D_p w_m(t) - C_{tm}[P_1(t) - P_2(t)] \quad (3.26)$$

Where β_e is the hydraulic fluid effective bulk modulus, C_{tm} is the leakage coefficient of the hydraulic motor. D_m is the volumetric displacement of the motor. w_m is the rotary speed of the hydraulic motor. V_1, V_2 are the original total control volumes of the hydraulic motor chambers. The others are the parts of the motor corresponding to the pump.

The torque balance equation of the hydraulic motor is formulated as

$$D_m(P_1(t) - P_2(t)) = J_t\ddot{\theta_m}(t) + G\theta_m(t) + T_p(t) \quad (3.27)$$

Where J_t and G are the total inertia of the hydraulic motor and pitch load spring gradient respectively. $T_p(t)$ represents the load torque which includes the active torque needed to drive the blades, which is converted through the pitch gear and resistance torque generated by friction increment on the hydraulic motor axis, which is treated as fault in this thesis.

The resulting pitch angle is described as

$$\beta = \frac{\theta_m(t)}{i_g} \tag{3.28}$$

Where β and i_g denote the pitch angle and gear ratio respectively.

The entire pitch angle system dynamics presented in (3.22) to (3.28) are expressed in (3.29).

$$\ddot{\theta} = \frac{1}{j_t}[D_m(\dot{P}_1 - \dot{P}_2) - G\ddot{\theta}_m - \dot{T}_p]$$

$$= \frac{D_m}{J_t}\begin{Bmatrix} \frac{\beta_e}{v_1 + D_m\theta_m}\left[D_p w_p - D_m w_m - (C_{tp} + C_{tm})(P_1 - P_2)\right] \\ -\frac{\beta_e}{v_2 - D_m\theta_m}\left[D_p w_m - D_p w_p + (C_{tp} - C_{tm})(P_1 - P_2)\right] \end{Bmatrix} - \frac{1}{J_t}(G\ddot{\theta}_m - \dot{T}_p) \tag{3.29}$$

A 3rd order state space model is set up using (3.29) by defining the state variables as

$x_1 = \beta(t), x_2 = \dot{\beta}, x_3 = \ddot{\beta}.$

$\dot{x}_1 = x_2$

$\dot{x}_2 = x_3$

$$\dot{x}_3 = \frac{D_m\beta_e}{J_t(v_1 + D_m i_g x_1)}\left[-D_m x_2 - \frac{(C_{tp} + C_{tm})}{D_m}\left(J_t x_3 + G x_1 + \frac{T_p(t)}{i_g}\right) + \frac{D_p K_a}{i_g}u(t)\right] \tag{3.30}$$

$$-\frac{D_m\beta_e}{J_t(v_2 - D_m i_g x_1)}\left[-D_p x_2 + \frac{(C_{tp} - C_{tm})}{D_m}\left(J_t x_3 + G x_1 + \frac{T_p(t)}{i_g}\right) - \frac{D_p K_a}{i_g}u(t)\right] - \frac{G}{J_t}x_3 - \frac{\dot{T}_p}{j_t i_g}$$

Where the leakage coefficient of the hydraulic pump and motor are denoted as C_{tp} and C_{tm} respectively.

The general form of (3.30) is formulated as follows

$$\dot{x}_3 = g(x) + b(x)u(t) + f(t) \tag{3.31}$$

$$g(x) = -\frac{D_m\beta_e}{J_t}\{\frac{1}{v_1 + D_m i_g x_1}\left[D_m x_2 + \frac{C_{tp} + C_{tm}}{D_m}(J_t x_3 + G x_1)\right]$$

$$+\frac{1}{v_2 - D_m i_g x_1}\left[D_m x_2 + \frac{C_{tp} + C_{tm}}{D_m}(J_t x_3 + G x_1)\right]\} - \frac{G}{J_t}x_3 \tag{3.32}$$

$$b(x) = \frac{D_m\beta_e D_p K_a}{J_t i_g}\left(\frac{1}{v_1 + D_m i_g x_1} + \frac{1}{v_2 - D_m i_g x_2}\right) \tag{3.33}$$

$$f(t) = -\frac{\beta_e(C_{tp}+C_{tm})}{J_t i_g} \times \frac{1}{v_1+D_m i_g x_1} \times T_p(t) - \frac{1}{J_t i_g}\dot{T_p}(t) \quad (3.34)$$

Where $f(t)$ represents the fault signal to be detected and tolerated in the control system designed. In this thesis $f(t)$ is a function of $T_p(t)$ and derivative of $\dot{T_p}(t)$. $T_p(t)$ is the added viscous and column friction of pitch system axis and gear set. This is caused by a frequently occurring system(component) failure in a wind turbine system. The physical parameters in the model are listed in Table 3.1.

Table 3.1: Physical parameters in the model [128]

Parameters	**Definitions**	**Values**
K_a	Gain of the servo motor unit	0.5 (rad/s/V)
D_p	Volumetric displacement of the pump	0.03 (m^3/rad)
C_{tp}	Leakage coefficient of the pump	$0.5*10^{-11}$ (m^3 /s/Pa)
β_e	Bulk modulus of the hydraulic fluid	$2.0*10^{-9}$(Pa)
V_1	Original total control volume of the hydraulic motor chamber in the high pressure side	0.01(m^3)
V_2	Original total control volume of the hydraulic motor chamber in the low pressure side	0.04(m^3)
D_m	Volumetric displacement of the motor	0.002 (m^3/rad)
C_{tm}	Leakage coefficient of the motor	$0.6*10^{-11}$(m^3/s/Pa)
J_t	Total inertia of hydraulic motor	138 (kgm^2)
G	Pitch load spring gradient	0.1 (kgm^2s^2)
i_g	Pitch angle gear ratio	30

3.7 Summary

In this chapter, the benchmark mathematical model of the wind turbine system was developed. The wind turbine systems are made up of aerodynamics, pitch actuator, drive train system and generator. The turbine is a variable speed and pitch controlled three-blade horizontal-axis turbine. A state space model was developed by the combination of the wind turbine blade pitch dynamics and drive train dynamics differential equations. A new electrohydraulic servo pitch system of a wind turbine was developed. It consists of a hydraulic motor, pitch gear set two relief valves and a variable speed hydraulic pump. The dynamic mathematical model and configuration of the pitch system was presented. The pitch angle of the blades is the controlled variables. A MATLAB and Simulink were used in this modelling.

CHAPTER 4

Fault Detection and Isolation (FDI) For Pitch and Drive Train Model Using Unknown Input Observer (UIO)

4.1 Introduction

The observer method is responsible for the generation of the plant's state estimate using the plant input and output signals [31]. In dynamic systems, a system can be subjected to unknown disturbances or unknown input. An unknown input observer has an important role in robust model-based fault detection and estimation [27]. In this section, a bank of unknown input observers (UIO) are modelled to generate a group of residuals achieving fault detection and isolation. The following state space model (4.01) represents a linear dynamic system,

$$\dot{x} = Ax + Bu + Ed(t) + Qf_a \tag{4.01}$$

$$y = Cx + Gf_s$$

Where $x \in R^n$ is the state vector, $y \in R^p$ is the output vector, $u \in R^m$ is the input vector, $f_a \in R^q$ is the actuator or component fault vector and $f_s \in R^r$ is the sensor fault vector. $A, B, Q, C \text{ and } G$ are known matrices with appropriate dimensions. The term Ed(t) in the system (4.01) can be described as disturbance as well as several kinds of uncertainties in modelling such as noise, linearization and model reduction errors and nonlinear terms in system dynamics.

4.2 Unknown input observer (UIO)

4.2.1 UIO structure

For the system above (4.01), a full order unknown input observer (UIO) is proposed with the structure described as (4.02).

$$\dot{z} = Fz(t) + TBu(t) + Ky(t)$$

$$\hat{x}(t) = z(t) + Hy(t) \tag{4.02}$$

where $\hat{x} \in \Re^n$ is the estimated state vector, $z \in \Re^n$ is the observer state, and F, T, K, H are matrices to be designed to achieve unknown actuator faults and input decoupling.

The estimation error (4.03) when the observer (4.02) is applied to the system (4.01) is governed by the following dynamics (4.04),

$$e(t) = x(t) - \hat{x}(t) \tag{4.03}$$

$$\dot{e} = \dot{x} - \dot{\hat{x}} = [Ax(t) + Bu(t) + Ed(t)] - [\dot{z}(t) + H\dot{y}(t)] \tag{4.04}$$

where $\dot{x}$ is replaced with (4.1) without the fault while $\dot{\hat{x}}$ is calculated by the derivative of (4.02). The error dynamics after substituting (4.02) to $\dot{z}$, (4.05), (4.04) and $\dot{y} = C\dot{x}$ in (4.04) is (4.07) The term Ed(t) in the system (4.04) can be described as disturbance.

(4.05) is derived from substituting $\hat{x} = z + Hy$ in (4.02) into (4.03).

$$x = e + z + Hy \tag{4.05}$$

$$K = K_1 + K_2 \tag{4.06}$$

$\dot{e} = (A - HCA - K_1C)e + [(A - HCA - K_1C) - F]z$ ...

$$+[(A - HCA - K_1C)H - K_2]y + (I - T - HC)Bu + (I - HC)Ed \tag{4.07}$$

(4.07) is considered if the following conditions are satisfied

$$(HC - I)E = 0 \tag{4.08}$$

$$T = I - HC \tag{4.09}$$

$$F = A - HCA - K_1C \tag{4.10}$$

$$K_2 = FH \tag{4.11}$$

The state estimation error will then be

$$\dot{e}(t) = Fe(t) \tag{4.12}$$

If all the eigenvalues of the matrix F are in the left-hand side of the s plane, the error will asymptotically approach zero, and the UIO estimation $\hat{x}$ asymptotically converges to x(t) in the presence of unknown input d(t).

4.2.2 UIO Design

The design of the observer is to find out the matrices H, K_1, K, F, T and K_2 to satisfy (4.08) to (4.11).

The tasks for designing it are listed as follows:

(i) Solve H using (4.08),

(ii) Use the pole placement method to select K_1 and obtain F to satisfy (4.10) and F is stable.

(iii) Calculate T using (4.09), K_2 with (4.11), and with (4.06).

4.2.2.1 The first step

(4.08) is $HCE = E$. Thus, H can be solved as $H = E(CE)^*$ if

$$rank(CE) = rank(E) \tag{4.13}$$

Where $(CE)^*$ is the left inverse of CE and is given as

$$(CE)^* = [(CE)^T CE]^{-1}(CE)^T \tag{4.14}$$

H has a general solution, which is presented in (4.15);

$$H = E(CE)^* + H_0[I_p - CE(CE)^*] \tag{4.15}$$

Where $H_0 \in \Re^{n \times p}$ is an arbitrary matrix.

4.2.2.2 The second step

(4.10) is of the form

$$\begin{aligned} F &= A - HCA - K_1 \\ &= A - \{E(CE)^* + H_0[I_P - CE(CE)^*]\}CA - K_1C \\ &= A - E(CE)^*CA - \{K_1C + H_0\left[I_p - CE(CE)^*\right]CA\} \\ &= [I_n - E(CE)^*C]A - [K_1 \;\; H_0]\begin{bmatrix} C \\ C[I_n - E(CE)^*C]A \end{bmatrix} \end{aligned} \tag{4.16}$$

Let

$$A_1 = [I_n - E(CE)^*C]A \tag{4.17}$$

$$\overline{K_1} = [K_1 \;\; H_0] \tag{4.18}$$

$$\overline{C_1} = \begin{bmatrix} C \\ CA_1 \end{bmatrix} \tag{4.19}$$

(4.16) then becomes,

$$F = A_1 - \overline{K_1}\,\overline{C_1} \tag{4.20}$$

(4.20) is a basic state observer design problem. $\overline{K_1}$ is solved using pole assignment method. The MATLAB function "place" in the "Control Systems Toolbox" can be used. If the desired poles are given in $P = [\lambda_1\ \lambda_2\ \ldots\ \lambda_5]$ for (4.20); $\overline{K_1}$ can be calculated with

$$\overline{K_1} = [place(A_1^T, C_1^T, P)]^T \tag{4.21}$$

Finally a stable observer matrix F is guaranteed to be achieved if the pair (C, A_1) and $(\overline{C_1}, A)$ is detectable.

The design summary is outlined below;

1. Test the two conditions, (4.13) and the pair (C, A_1) is detectable. If satisfied then proceed; otherwise, the UIO does not exist.
2. Determine the desired poles and calculate $\overline{K_1}$ with (4.21),
3. Determine K_1 and H_0 with (4.18) and calculate F with (4.20),
4. Calculate H with (4.15),K_2 with (4.11), and T with (4.09).

4.3 FDI for wind turbine

4.3.1 Wind turbine model without fault

The wind turbine parameters used in [1] are stated in table 4.1. These parameters are substituted into (4.01). Since there is no fault considered here E, Q and G are all zero matric.

Table 4.1: Wind turbine parameters [1]

Parameters	Value
N	97

J_r	5.9154×10^7
J_g	500
D_g	6.215×10^6
K_s	8.6763×10^8
ζ	0.9
R	63
ρ	1.225
τ	0.1
ω_n	0.88

The other matrices A, B and C are calculated on mat lab using the parameters in table 4.01. These values are stated below.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & -0 & 0 & 15 & 0 \\ 0 & 128 & -1 & 17889 & 0 \\ 0 & 1 & -0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -2 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 1.6905 * 10^{-8} & 0 & 0 \\ 0 & -0.0020 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.7744 \end{bmatrix}$$

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

The system input is chosen as

$$V_w(t) = \begin{cases} 12, & 0 < t < 50 \\ 15, & 50 < t < 100 \end{cases} \quad (4.22)$$

$$\beta_{ref} = 35 + \sin(0.04 * pi * t) \quad (4.23)$$

$$T_{gref}(t) = \frac{P_r}{w_{rated}*N} \quad (4.24)$$

4.3.2 Fault Simulation

Different kinds of faults were considered in this thesis. These faults include actuator and sensor faults.

4.3.2.1 Actuator fault

The actuator fault is denoted as $\Delta\tau_g$. The cause of this fault is due to the malfunction of some components in the converter [3]. The fault is assumed to occur on the generator torque T_g between 10 and 20 secs. The value of T_g increased by 70%. The consequence of this fault is that the torque control is slow [3]. The system experienced an unknown load disturbance in the pitch angle of the rotor in the time frame 40 to 50 secs. The disturbance $d(t)$ is chosen as

$$d(t) = \begin{cases} 1000, & 0 < t < 40 \\ 200, & 40 < t < 50 \\ 1000 & 50 < t < 100 \end{cases}$$

The M file is then run to simulate the faults.

4.3.2.2 Sensor Fault

Here three different sensor faults are considered. The pitch angle measurement is the first sensor fault to be considered. The fault is assumed to occur during the test time 20 to 30 secs. The generator speed sensor is the second sensor fault to be simulated. There was a 50% drop in the measured rotor during time 40 to 50 secs. The last sensor fault to be discussed here is the measurement of rotor speed. The wrong value was shown between 30 and 40 secs.

4.3.3 WT model with fault

According to the simulated faults and disturbance the matrices E, Q and G are constructed on matlab and the values are stated below.

$$Q = \begin{bmatrix} 0 \\ 0 \\ -0.0020 \\ 0 \\ 0 \end{bmatrix}, \qquad E = \begin{bmatrix} 0 \\ 1.6905 * 10^{-8} \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad G = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

4.3.4 UIO design for WT

The UIO design parameters H, K_1, F, T and K_2 could be calculated to satisfy (4.08) to (4.11). This is calculated on Matlab and the values of the matrices are stated below.

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -0.5 & 0 \\ 0 & 0 & 0 & 0.0670 & 0 \\ 0 & 0 & -0.0001 & 0.5071 & 0 \\ -2.4160 & 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & -0.0670 \\ 0 & 0 & 0.0001 & 0.4929 \\ 2.4160 & 0 & 0 & 0 \end{bmatrix},$$

$$K_1 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 0.5 & 0.0052 & -0.0036 \\ 0 & 0.0670 & 12.9993 & 0.0005 \\ 0 & 0.4928 & -0.0051 & -0.0035 \\ -0.7744 & 0 & 0 & 0 \end{bmatrix} K_2 = \begin{bmatrix} -10 & 0 & 0 & 0 \\ 0 & -1 & 0 & -0.5 \\ 0 & 0 & -13 & 0.8711 \\ 0 & 0 & 0.0002 & -0.9857 \\ -9.6640 & 0 & 0 & 0 \end{bmatrix},$$

$$F = \begin{bmatrix} -10 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -13 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -4 \end{bmatrix}$$

4.4 SIMULINK STUDIES

4.4.1 Simulation model

The Simulink model is designed with three sub-models, one for the system and the other two for the aerodynamic torque and UIO respectively. The UIO is presented in Figure 4.04 The plant is built according to (4.01) and can be shown in Figure 4.01. The aerodynamic torque is designed using (3.01) and presented in Figure 4.05.

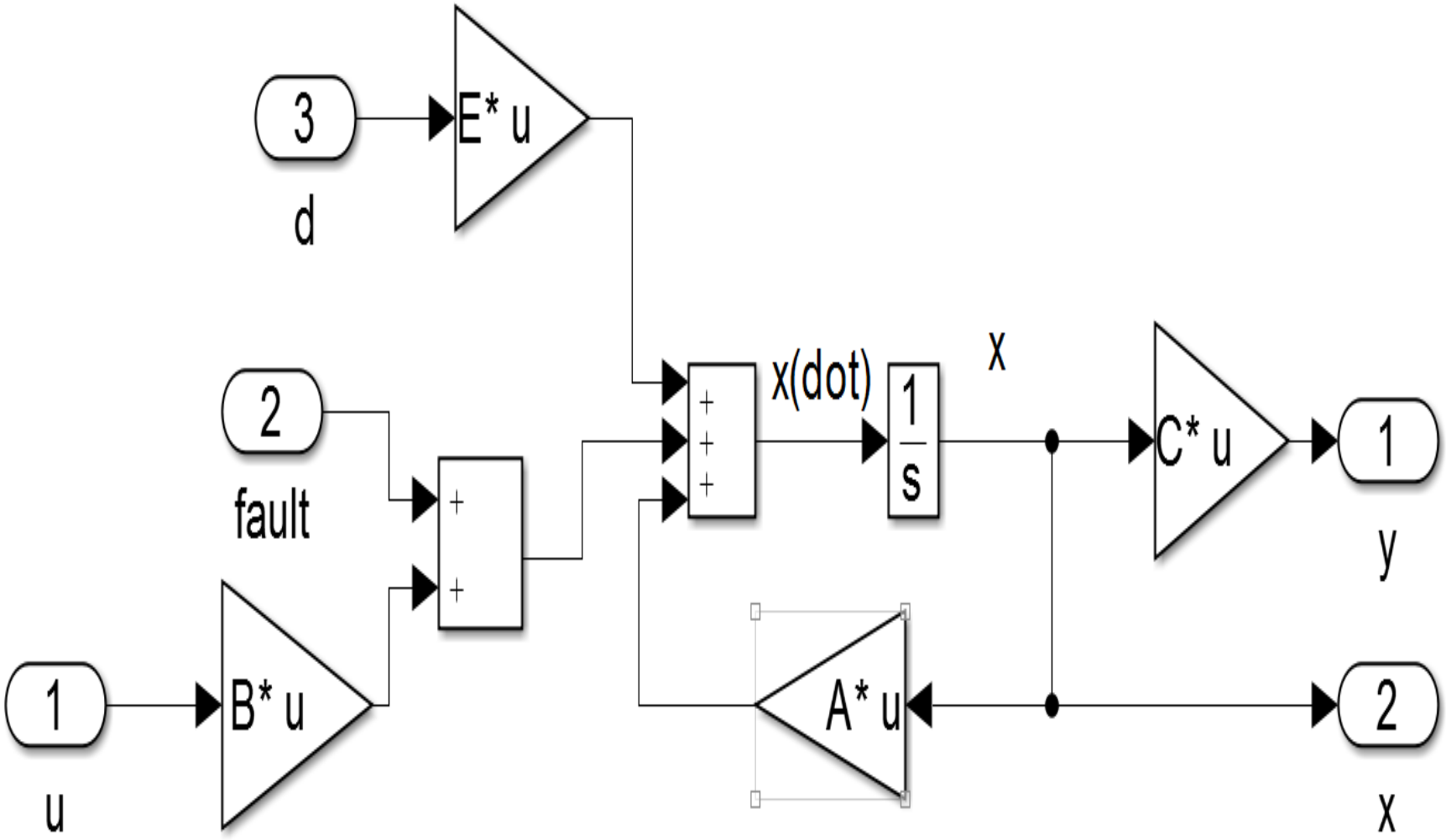

Figure 4.01: The wind turbine plant

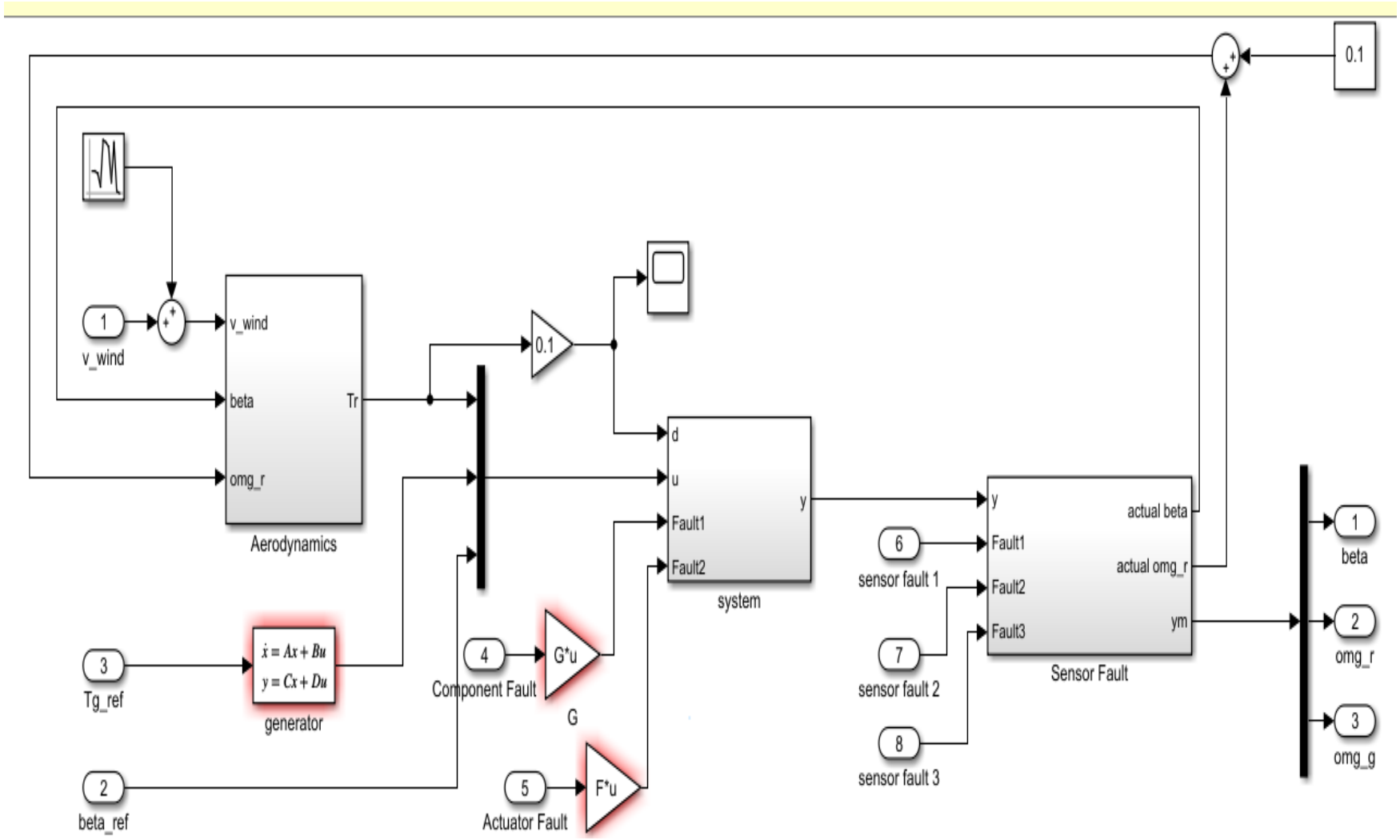

Figure 4.02: WT model with faults

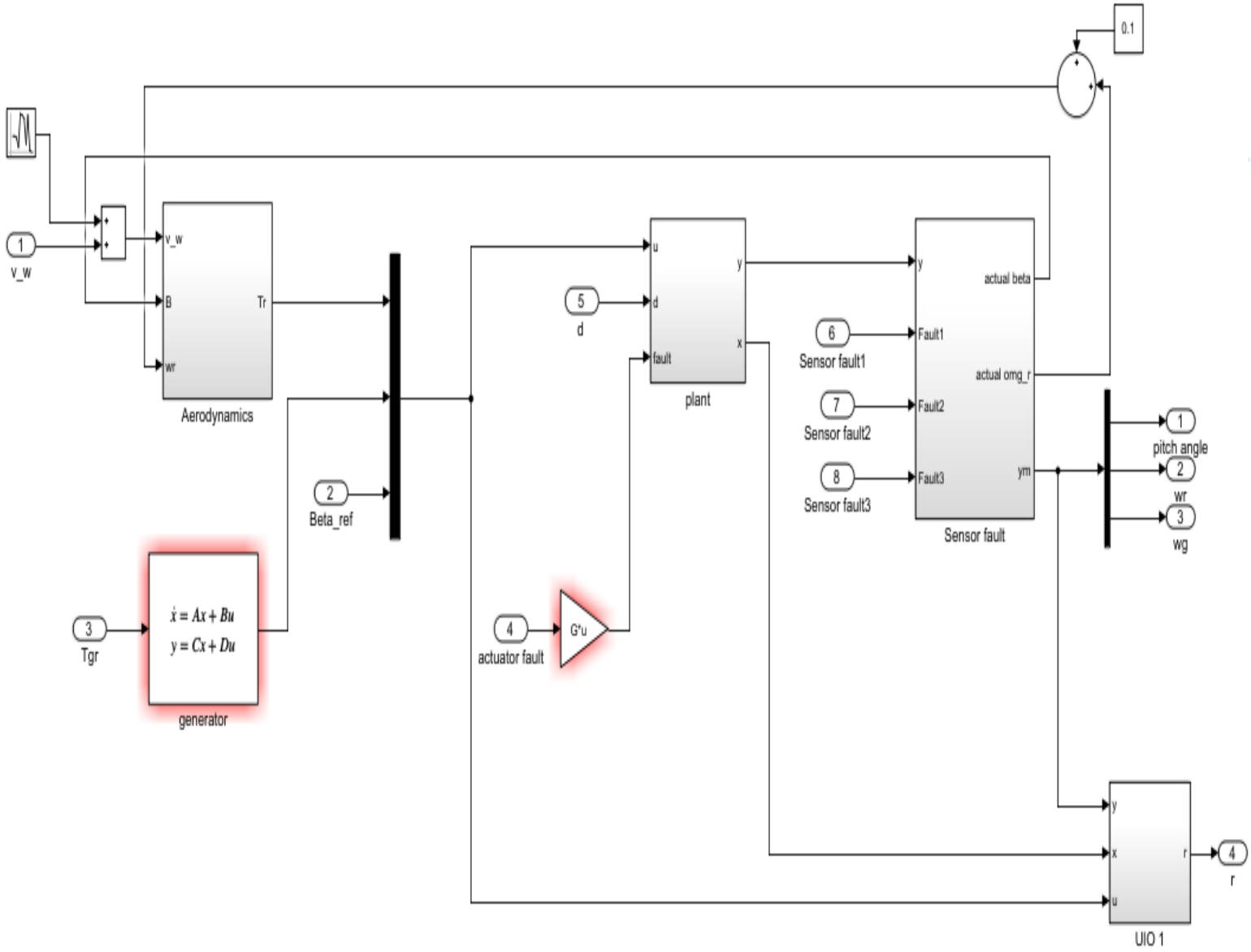

Figure 4.03: Fault Detection model

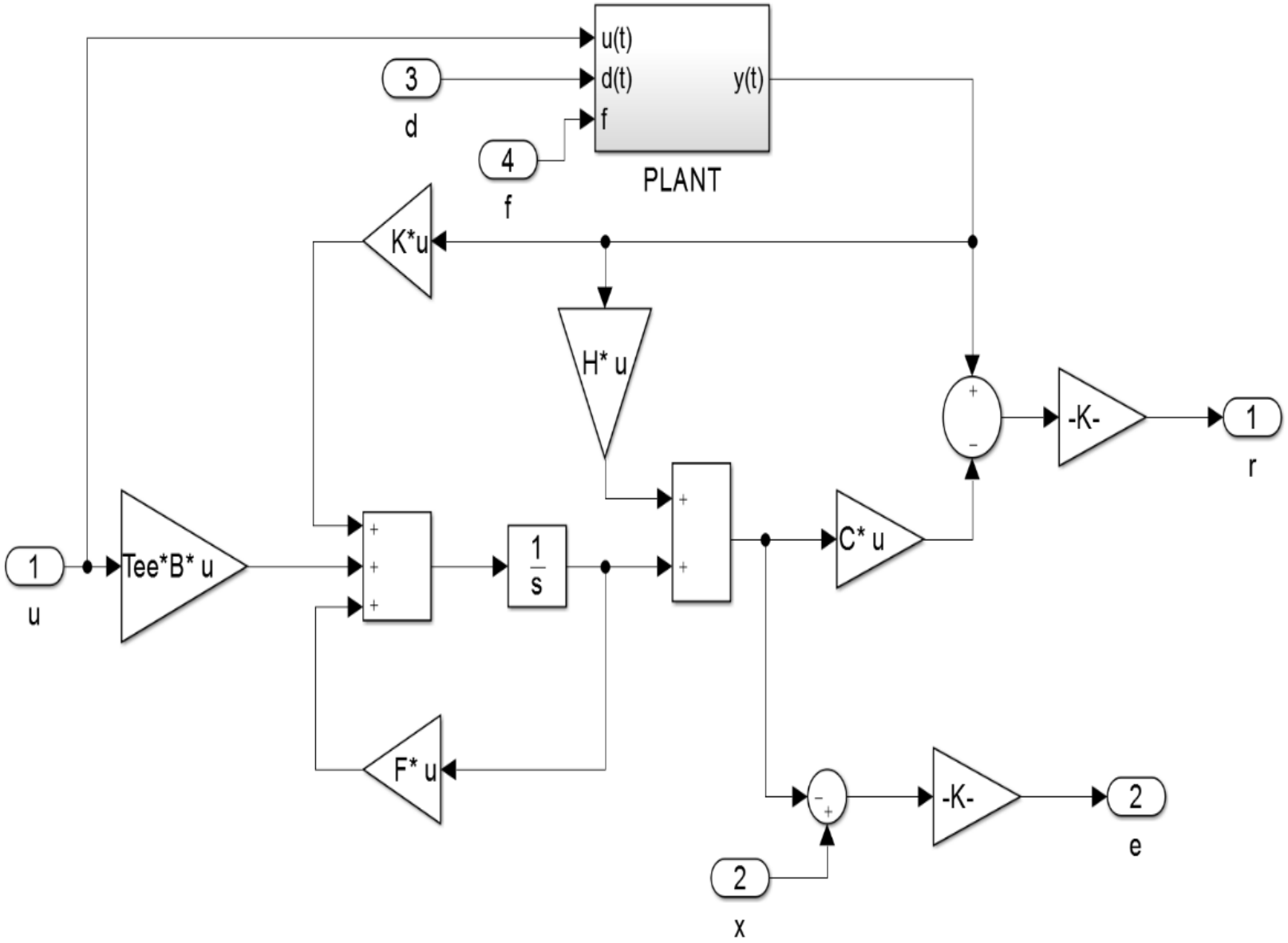

Figure 4.04: Sub-model 1: Structure of the full-order UIO

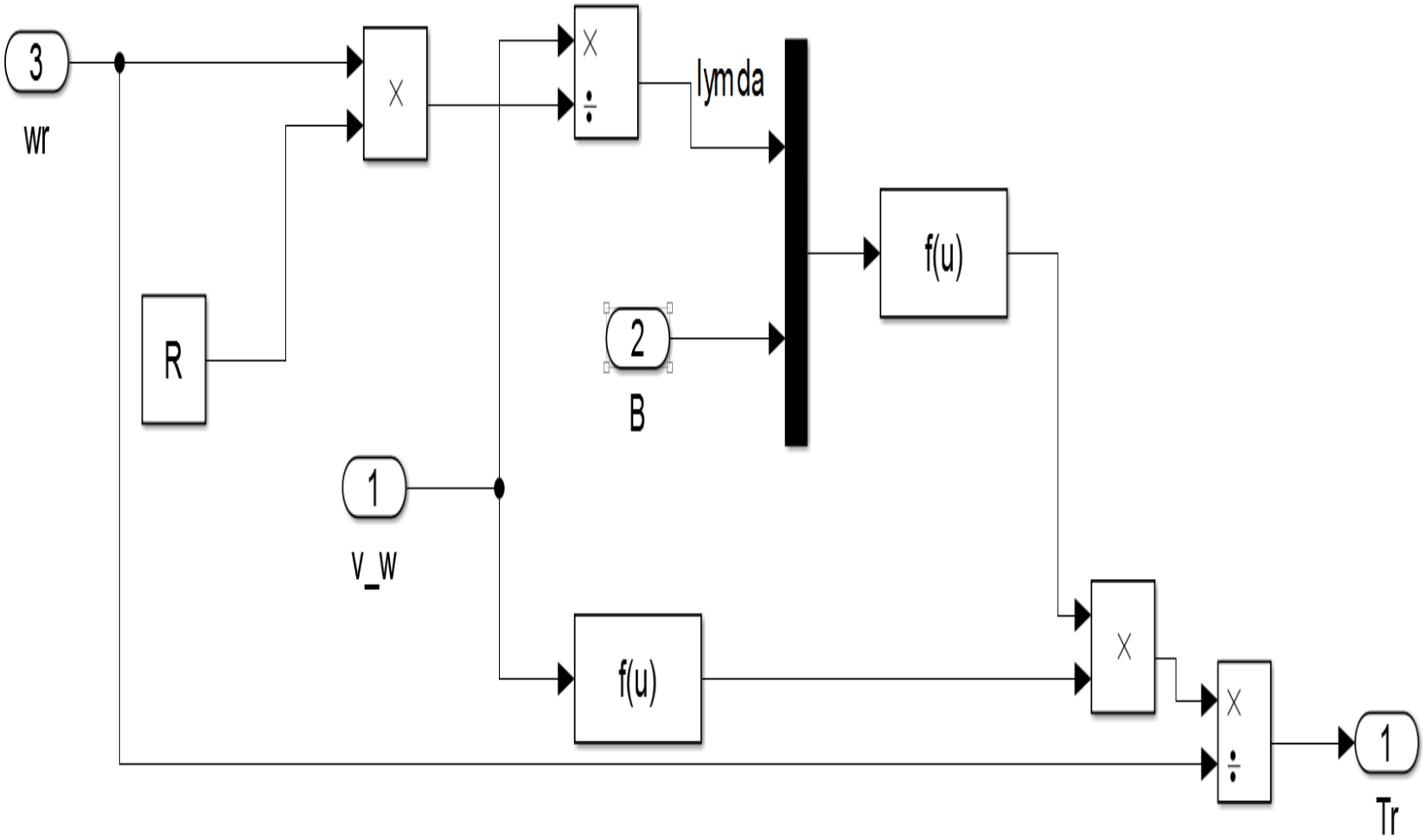

Figure 4.05: Sub-model 111: The aerodynamic torque

4.4.2 Simulink results

4.4.2.1 No fault

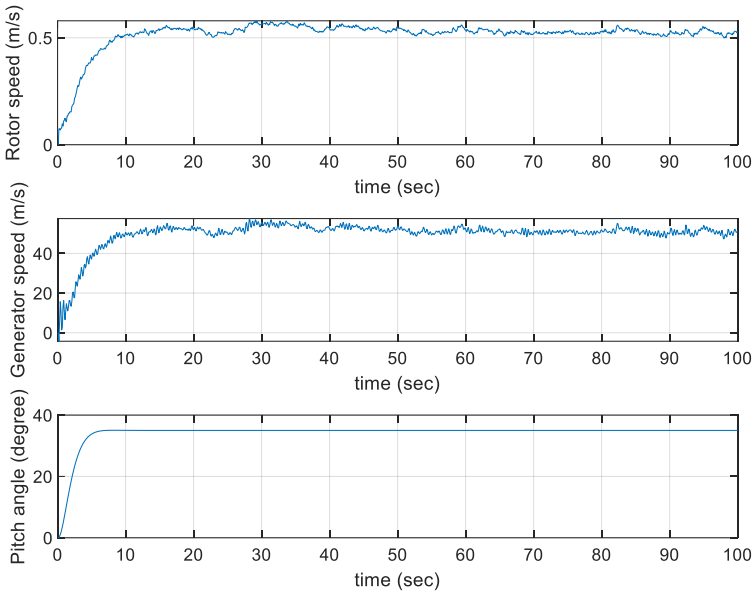

Figure 4.06: Wind turbine output with no fault.

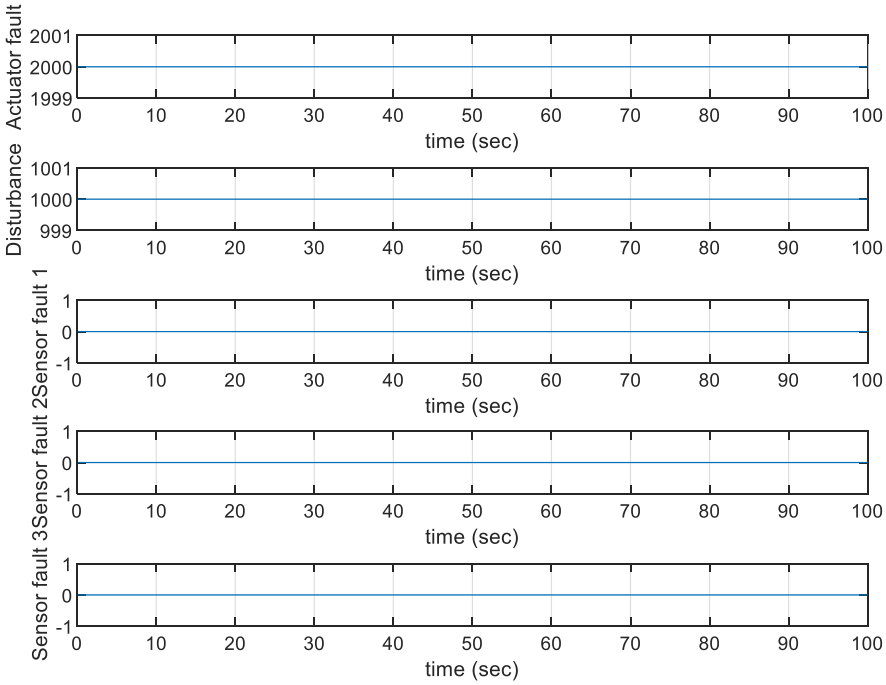

Figure 4.07: Changes of Residual during no fault and no disturbance.

4.4.2.2 Actuator fault

Figure 4.08 displays the changes of residual signal during actuator fault including the disturbance. A close look at Figure 4.08 reveals that the residual signals are non-zero within the time frame of 110 - 120 seconds. It is also observed that the system is robust to the disturbance on the rotor torque within the time frame 140 to 150 secs.

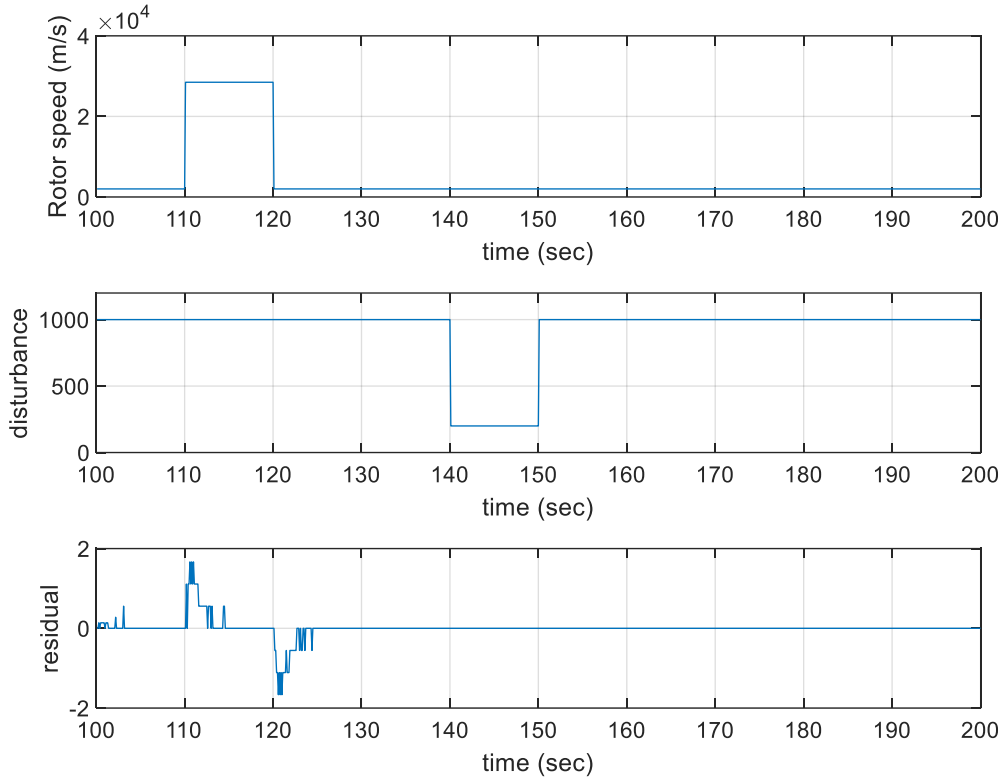

Figure 4.08: Changes of Residual during actuator fault and disturbance.

4.4.2.3 Sensor faults

In this research three different sensor faults are considered. The pitch angle measurement is the first sensor fault to be considered. We assumed that the fault in the pitch angle sensor occurred during the test time 220 secs to 230secs. This is shown in figure 4.09 where the measured pitch angle changed to 70 degrees and the residual also deviated from zero at the fault time.

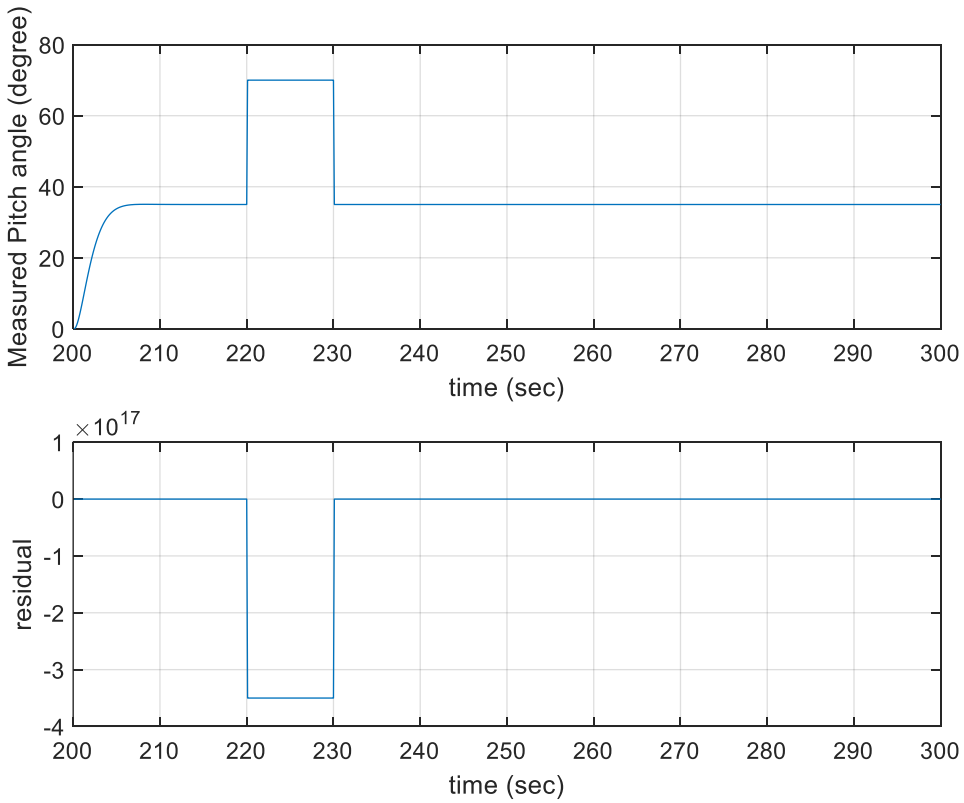

Figure 4.09: Changes of Residual during pitch angle sensor fault.

The rotor speed sensor is the second sensor fault considered in this research. The measured rotor speed value increased by 0.25 during the fault times 330 secs to 340 secs. The residual deviates from zero when the fault has occurred as presented in figure 4.10. Figure 4.11 shows the generator speed sensor fault and the residual. The value for the generator speed was incorrect between 440 secs and 450secs. The residual also deviated from zero at the fault time.

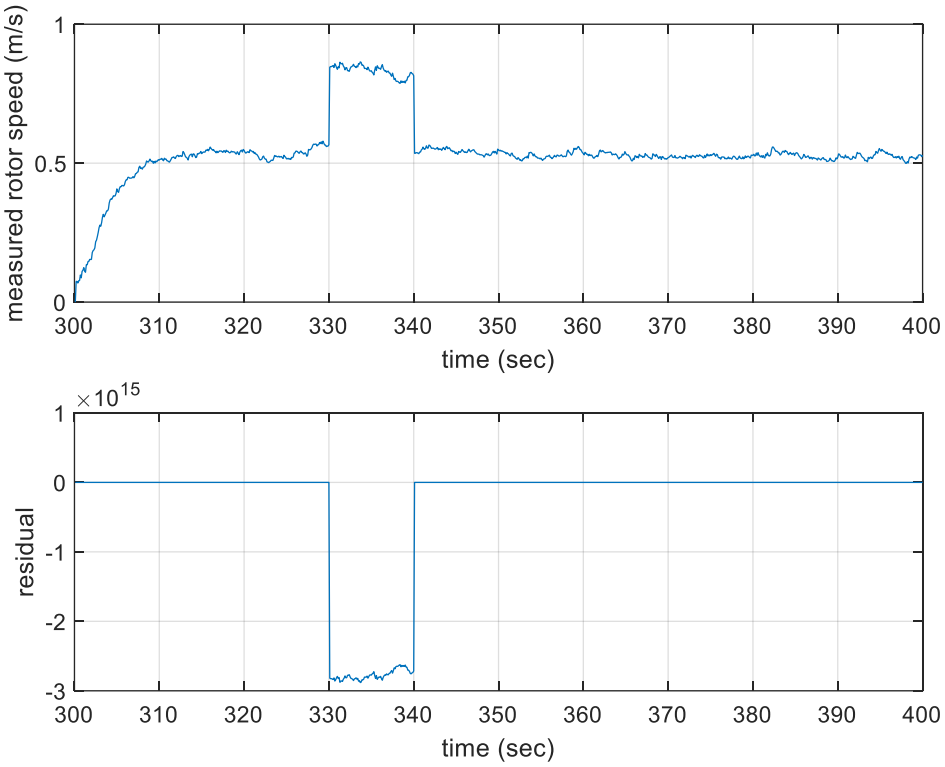

Figure 4.10: Changes of Residual during rotor speed sensor fault

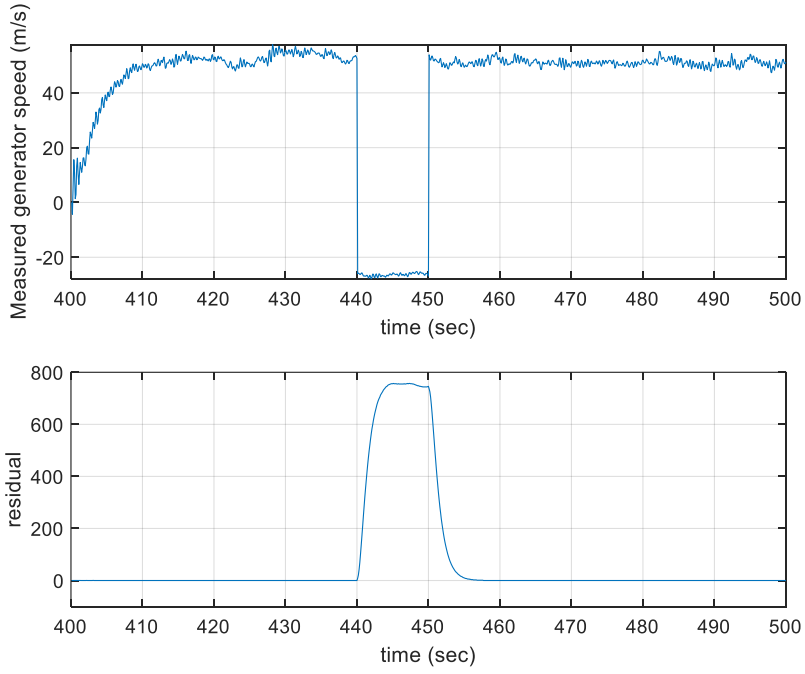

Figure 4.11: Changes of Residual during rotor speed sensor fault.

4.5 Summary

A robust fault detection approach using unknown input observer is proposed. A robust UIO which is sensitive to faults in the state equation and output equation while robust to system disturbance is designed using the wind turbine system. Actuator fault and three sensor faults were simulated, and the results show that the observer is working as desired. The fault detection residual is robust to disturbances and sensitive to simulated faults. This method was used to check if the fault in the linear model of the wind turbine can be simulated.

CHAPTER 5

Fault Diagnosis for Wind Turbine Systems Using A Neural Network Estimator.

5.1 Introduction

A new robust method is developed for fault detection and isolation of a wind turbine system. The fault diagnosis method engages a RBFNN in an independent mode to diagnose faults in a wind turbine using the weighted sum-squared prediction error as the residual. Radial basis function neural network (RBFNN) is chosen over the unknown input observer developed earlier because of its quality to approximate a nonlinear input system to a linear output [100]. For the fault isolation another RBFNN is used as a fault classifier to isolate faults in the residual vector. Different actuator fault and sensor faults are simulated in the wind turbine system on Simulink. The robust method is used to detect and isolate faults in the wind turbine system. When compared with other neural networks, the training process of RBFNN is fast and better [125].

5.2 Radial basis function (RBF) network configuration and training algorithms

The RBF network is a feed-forward network made up of three layers: input layer, hidden layers and output layer where $x = [x_1, x_2, x_3 \ldots . x_m]^T \in R^m$ is the input vector, $h = \left[h_1, h_2, h_3, \ldots h_q\right]^T \in R^q$ is the output vector of the hidden layer [33]. $w(k) \epsilon R^{p \times q}$ is the weight of the matrix with entry w_{xy}, which is simply the weight linking the yth node in the hidden layer to the xth node in the output layer and the output vector of the RBF network is represented as $\hat{y} = \left[\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_q\right]^T$. Figure 5.01 shows the structure of the RBFNN.

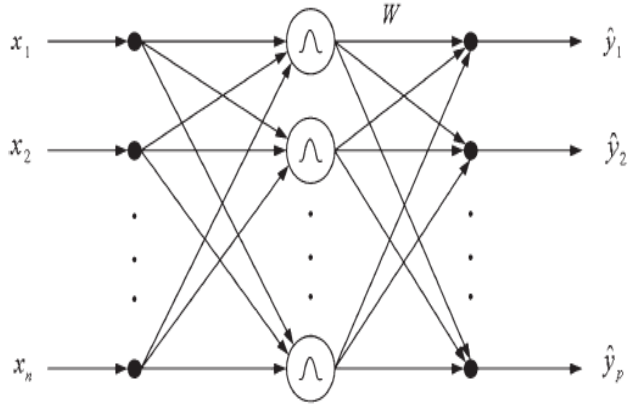

Figure 5.01: RBFNN Structure

The RBF is represented in mathematical terms as

$$\hat{y}(k) = \frac{w}{h(k)} \tag{5.01}$$

$$h(k) = f[z(k)] \tag{5.02}$$

$$z_i(k) = \sqrt{[x(k) - C_i]^T [x(k) - C_i]} = \|x(k) - C_i\| \tag{5.03}$$

where $i = 1,2,\cdots q$. $f[\cdot]$ is the nonlinear activation function in the hidden layer. $[\cdot]$ represents the equation for finding the Gaussian basis function.

$$f[z(k), \sigma] = e^{-\left(\frac{z^2(k)}{\sigma^2}\right)} \tag{5.04}$$

where σ is a positive scalar known as width, which is defined as a distance scaling parameter to determine the distance in the input space over which the unit will have a significant output.

In this research the RBF network will be used to predict some unmeasurable variables. This is achieved by the following steps.

- determine network input according to the dynamics of the system,
- data collection and scaling,
- network training and validation.

In the training of the network the number q of centers is determined, the appropriate centres C_i and widths $\sigma_i, i = 1,2,3,\ldots q$ is found from the training data sets and the weight w is obtained by training data and validating the network in healthy operating conditions. The training algorithms are explained briefly.

5.2.1 K-means algorithm

In this research, the centres of the RBF network are chosen using the algorithm from a set of training data. This is achieved using the K-means clustering algorithm. Its objective is to minimize the sum squared distances from each input data to the closest centre to ensure adequately covered data by the activation function $f[\cdot]$. The steps are listed below;

- Choose q initial cluster centres $c_1(1), c_2(1), \ldots c_q(1)$.
- Distribute the sample {x} into $S_j(t)$ among the q clusters domain at the i^{th} iteration step. The set of samples whose cluster is $C_j(t)$ are denoted as $S_j(t)$.

$$x \epsilon S_j(i)\ if\ \|x(k) - C_i\| < \ \|x(k) - C_i\|$$

Where $j = 1,2,3,\ldots q, i = 1,2,3,\ldots q$.

- Update the cluster centres

$$c_j(i+1) = \frac{1}{N_j}\sum_{j}^{N_j} S_j(t)$$

Where N_j is the number of elements in $S_j(t)$

- Repeat the second and third step above until

$$C_j(i+1) = c_j(t)$$

5.2.2 P-nearest neighbour method

The p nearest neigbours method is used in computing the RBF network width σ of each unit. The procedure is that the excitation of each node should overlap with some other nodes (mostly the closest) to ensure the smooth interpolation surface between nodes to be obtained. For this to be achieved each hidden node must activate at least one other hidden node to a significant degree. The width is selected so that σ is greater than the distance to the nearest unit centre. This can be calculated using (5.05).

$$\sigma_i = \left(\frac{1}{p}\sum_{j=1}^{p}\left\|C_i - C_j\right\|^2\right)^{\frac{1}{2}} \quad (5.05)$$

Where $i = 1,2,3, \ldots q$ and C_j is the p-nearest neighbour of C_i. For the nonlinear approximation,p depends on the problem and requires experiments to be made to find it.

5.2.3 Recursive least square algorithm

This is a recursive form of the least squares algorithm. It is used here to determine the RBF network weights W which is summarized as follows [32]

$$Y_p(t) = Y_c(t) - W(t-1)h(t) \quad (5.06)$$

$$g_z(t) = \frac{P_{z(t-1)h(t)}}{\mu + h^T(t)P_z(t-1)h(t)} \quad (5.07)$$

$$P_z(t) = \mu^{-1}[P_z(t-1) - g_z(t)h^T(t)P_z(t-1)] \quad (5.08)$$

$$W(t) = W(t-1) + g_z(t)Y_p(t) \quad (5.09)$$

Where $W(t)$ and h(k) represents the RBF network weights and activation function outputs respectively at iteration k, $y_c(k)$ is the process output vector, and P_z and g_z are middle terms. μ here

is called the forgetting factor ranging from 0 to 1 and is chosen to be 1 for offline training. The parameters $g_z, W\ and\ P_z$ are updated orderly for each sample with change in the activation function output h(k).

5.3 FDI for wind turbines using RBF network.

We will detail the fault diagnosis of a wind turbine using an RBFNN. At a healthy condition an RBFNN is trained with data collected from the wind turbine at no fault. The model is placed parallel to the wind turbine in an online mode to predict the output of the wind turbine. The residual signal is the modelling error between the wind turbine output and the predicted model. This indicates that when there is no fault in the wind turbine the residual displays the modelling error caused by noise. When a fault occurs the wind turbine output will be affected, and the values will deviate from the nominal values while the predicted model is not affected by the fault. The residual will be able to pick the fault because there will be a significant deviation from zero. The importance of fault isolation in the FDI process is the capacity to classify the time and type of faults that occurred. An additional RBF NN is used as a classifier to achieve the fault isolation. The importance of an additional RBF NN is to build a framework of nonlinear observer for isolation of more than 3 faults. The information of all faults is included in the modelling error. The modelling error will be used as the input to the classifier as different faults will affect each element in the modelling error vector differently. The training of the RBF NN is outlined below; collect engine data for each of the faults occurring, these data are fed into the classifier with the non-fault target being all zeros (0) and the output of the fault indicated as ones (1). In an on-line mode, any output which turns to 1 shows that the faults associated with it occurs. If all the faults occur differently and have different characteristics the faults can be isolated clearly with this characteristic. Figure 5.02 shows Schematic of fault diagnosis for a WT using RBF networks.

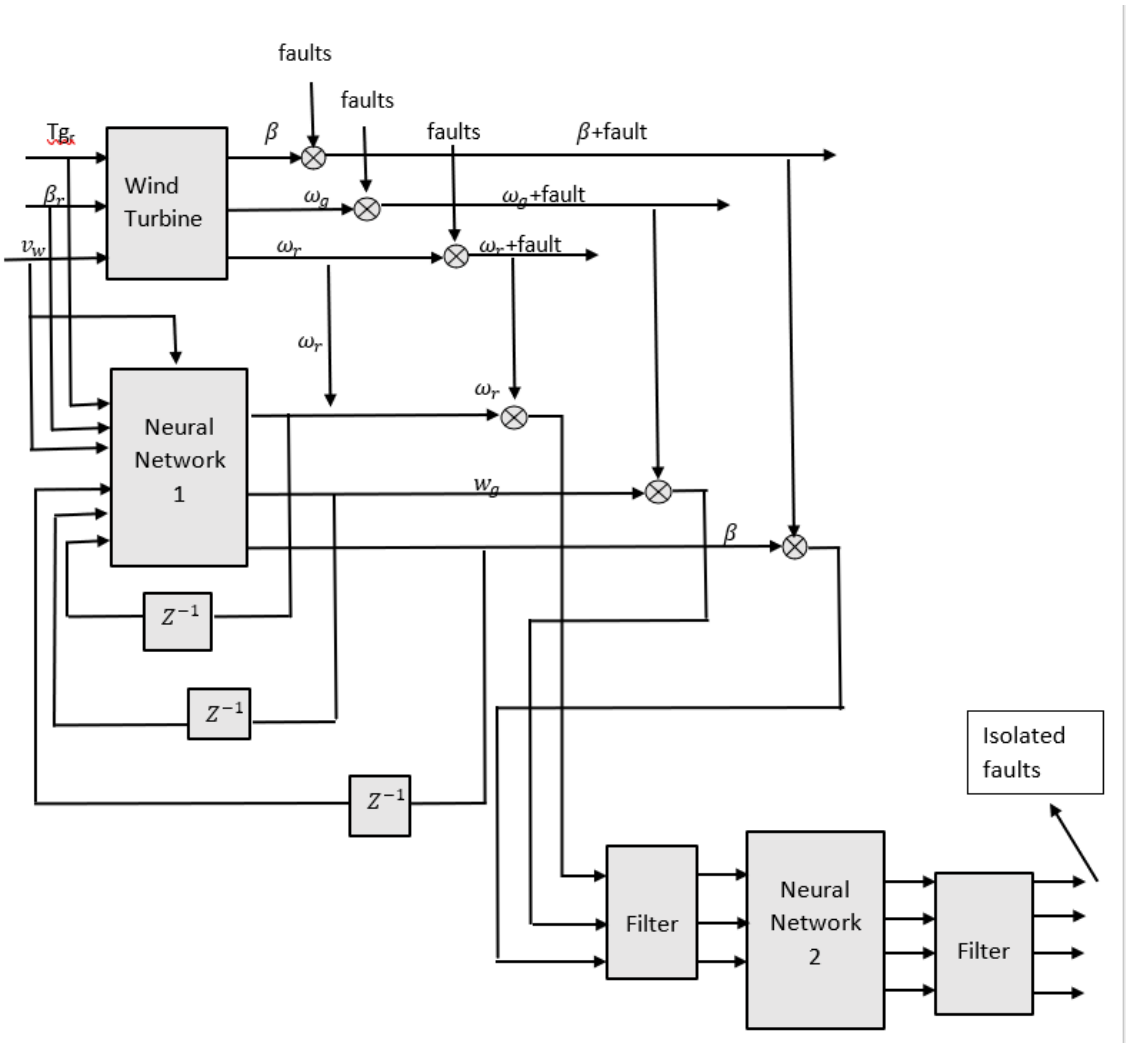

Figure 5.02: Schematic of fault diagnosis for engines using RBF networks.

5.4 RBF Model of a Wind turbine System

5.4.1 Data collection

This is the first step towards developing a RBFNN. Here a set of random amplitude signals (RAS) are generated and used as wind turbine system input signals. The three input signals are the wind speed, reference blade pitch angle and reference generator torque. The ranges of these signals are presented in table 5.1. This is generated randomly to cover the whole range of frequencies and entire operation space of amplitude in the wind turbine.

Table 5.1: Wind turbine input signals

RAS	Minimum	Maximum
B_{ref}	10	35
T_{gr}	4068	40680
v_w	5	20

All the raw data samples are scaled into the range [0, 1] using (5.10) to increase the accuracy of the RBF and decrease the error.

$$U_{scale} = \frac{u(k)-u_{min}}{u_{max}-u_{min}}$$

$$y_{scale} = \frac{y(k)-y_{min}}{y_{max}-y_{min}} \quad (5.10)$$

Where $u_{\min,} u_{\max,} y_{\min,} y_{\max,}$ are the minimum and maximum inputs-outputs of the data sets, while U_{scale} and y_{scale} are the scaled input and outputs respectively. The random amplitude signals (RAS) of the three inputs are shown in Figures 5.03 to 5.05.

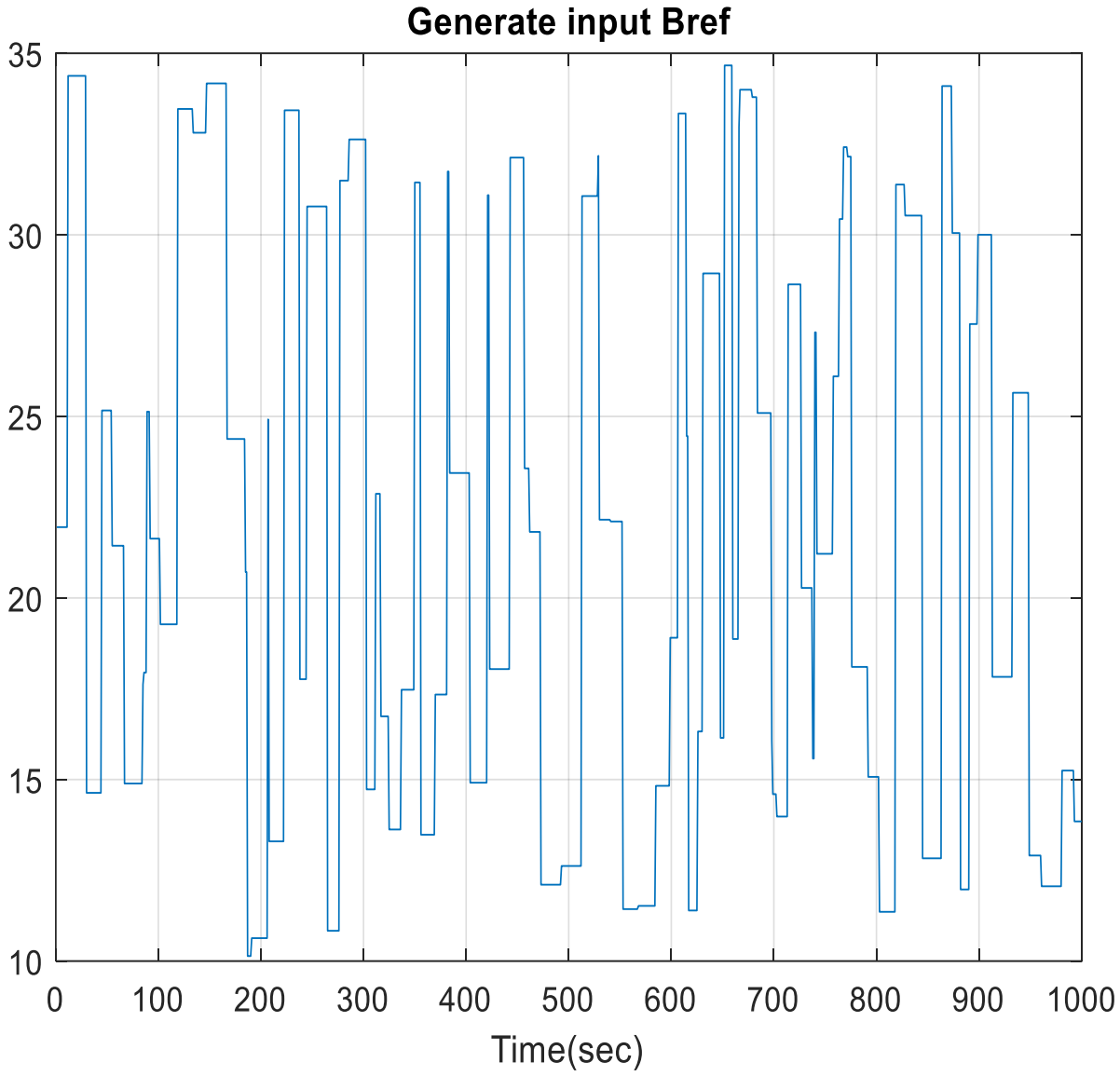

Figure 5.03: Random amplitude signal of pitch angle reference

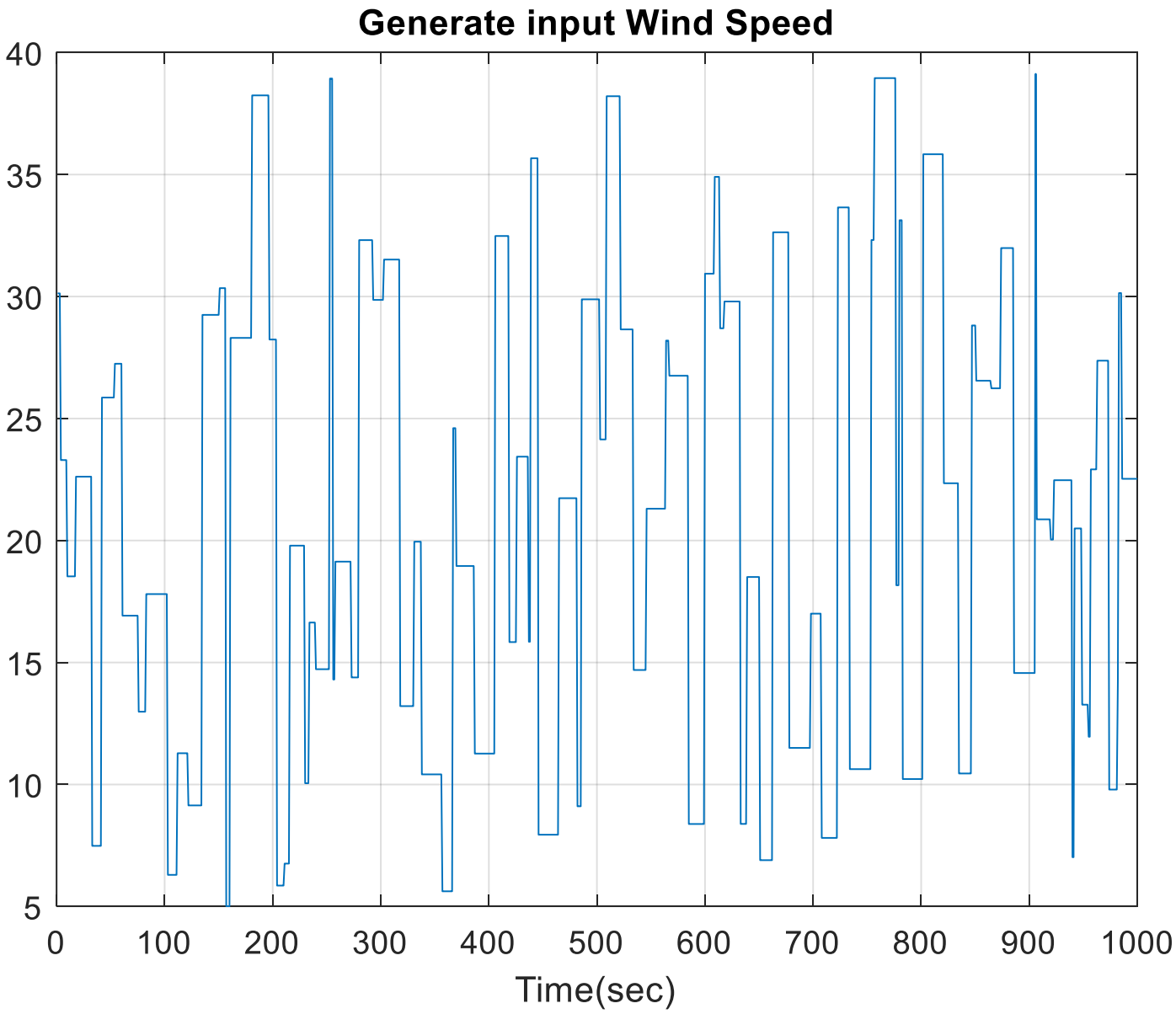

Figure 5.04: Random amplitude signal of wind speed

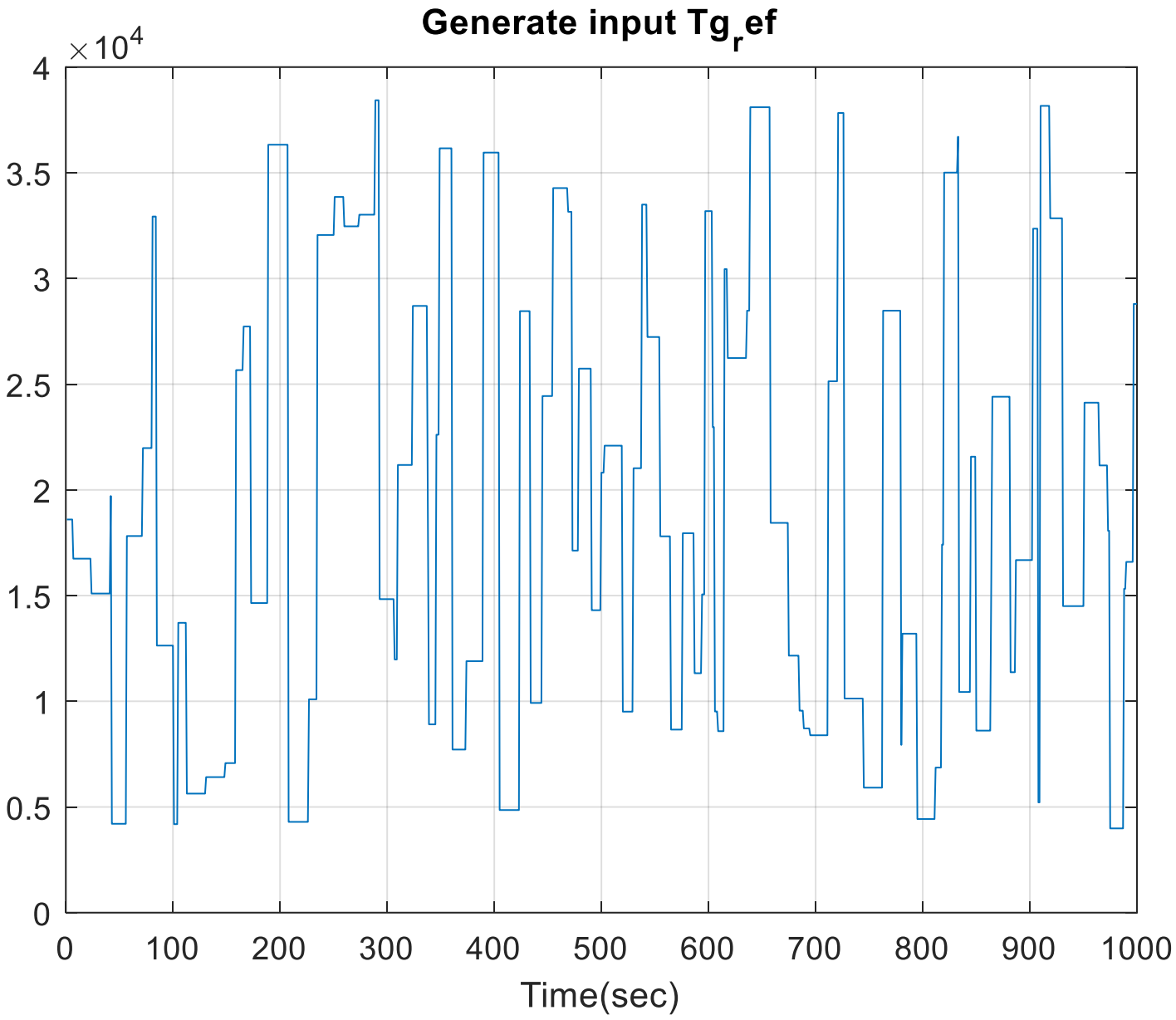

Figure 5.05: Random amplitude signal of reference generator torque

Figure 5.06 the wind turbine Simulink model is presented with three inputs including the wind speed. A random number was used to generate the wind speed as it is difficult to measure the wind speed in real life. The set of RAS generated in Figures 5.03 to 5.05 are used as the inputs of the wind turbine Simulink model. This can be shown in Figure 5.07.

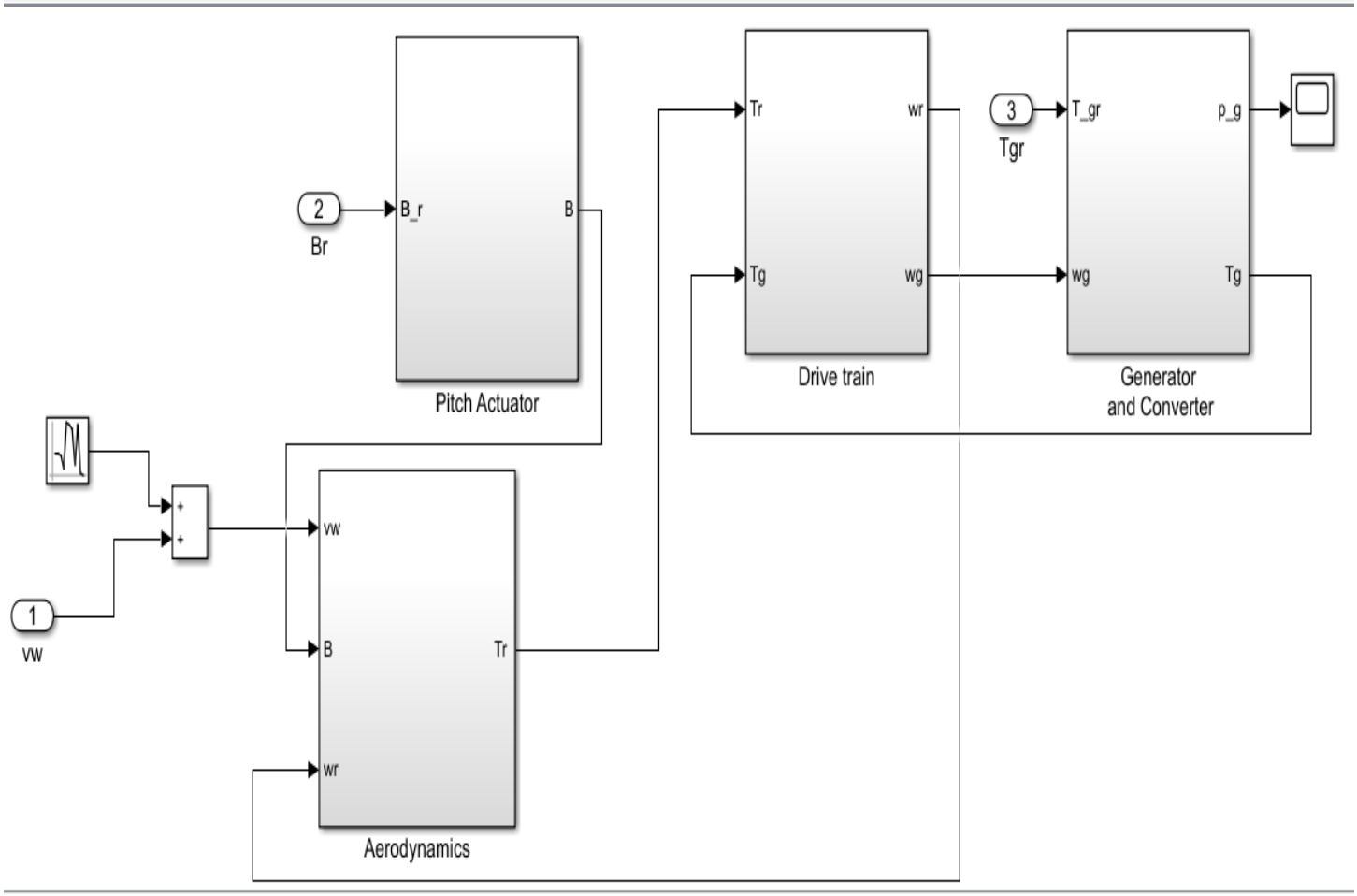

Figure 5.06: Wind turbine Simulink model

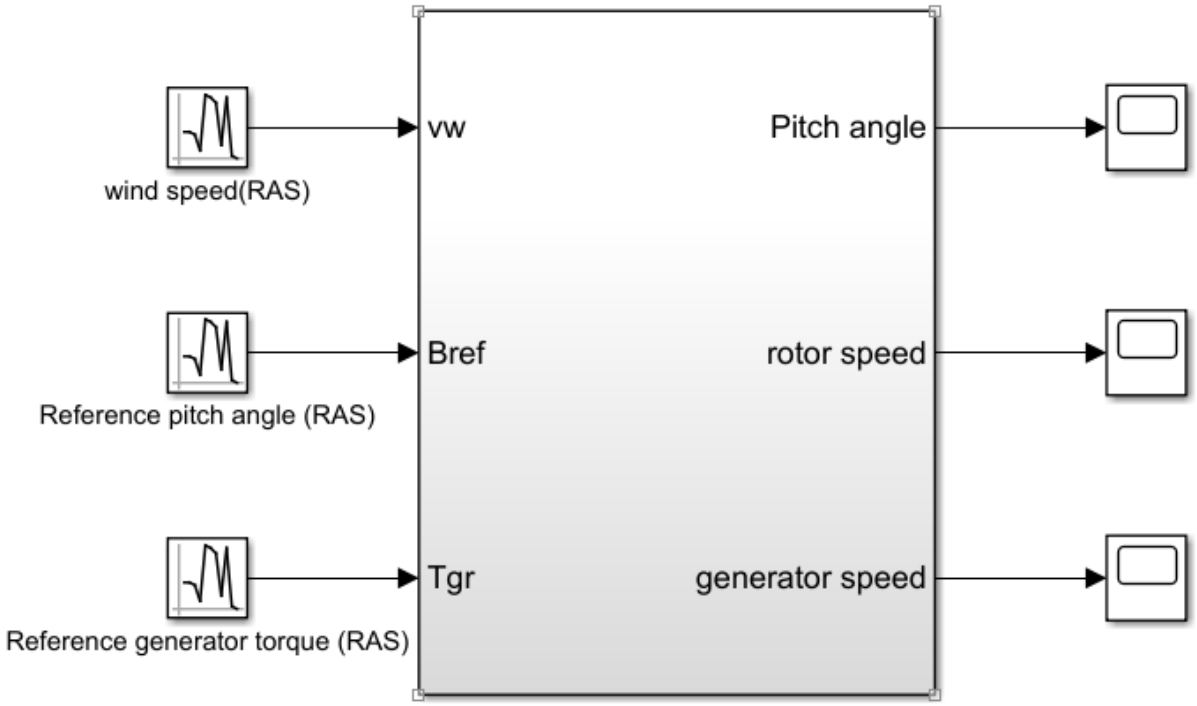

Figure 5.07: Simulink model of wind turbine with RAS input

5.4.2 Modelling structure

The input of the RBF model will now be determined. The selection is based on modelling trials where the model structure generates the smallest modelling errors. The equation is

$$\hat{y}(k) = f[\hat{y}(k-1)\hat{y}(k-2)\hat{y}(k-3), u(k-1), u(k-2), u(k-3)] + e(k)$$

The selected structure is made up of 18 inputs and 3 outputs. The hidden layer was chosen to be 8. This is chosen by experiment. The center is calculated using k-means clustering algorithm and the width σ was chosen using the p-nearest neighbors. The training algorithm used in [32] was used. $\mu = 0.99$, $w(0) = 1 \times 10^{-8} \times U_{nh\times 3}$,$P(0) = 1 \times 10^{8} \times I_{nh}$ where I is an identity matrix ,n_h is the number of hidden layers while μ is the forgetting factor. The simulation will run for 1000secs with 1000 data samples collected. These data samples were used to train the RBF NN. Figures 5.08 to 5.10 show the model training and validation results of pitch angle, rotor speed and generator speed respectively. This shows a good match between the wind turbine output and the RBF model output with small error. 8 centers gave minimum prediction error (MPE) as shown in table 5.2. The simulation results of training and testing the RBF NN model using 8 hidden nodes at nominal

condition were very good and a good prediction between the wind turbine output and RBF NN output was achieved. The mean absolute error index is used to assess the modelling effects.

Table 5.2: Minimum prediction error (MPE)

Outputs	Minimum Absolute Error (MAE)
Pitch angle	0.0286
Rotor speed	0.0181
Generator speed	0.0181

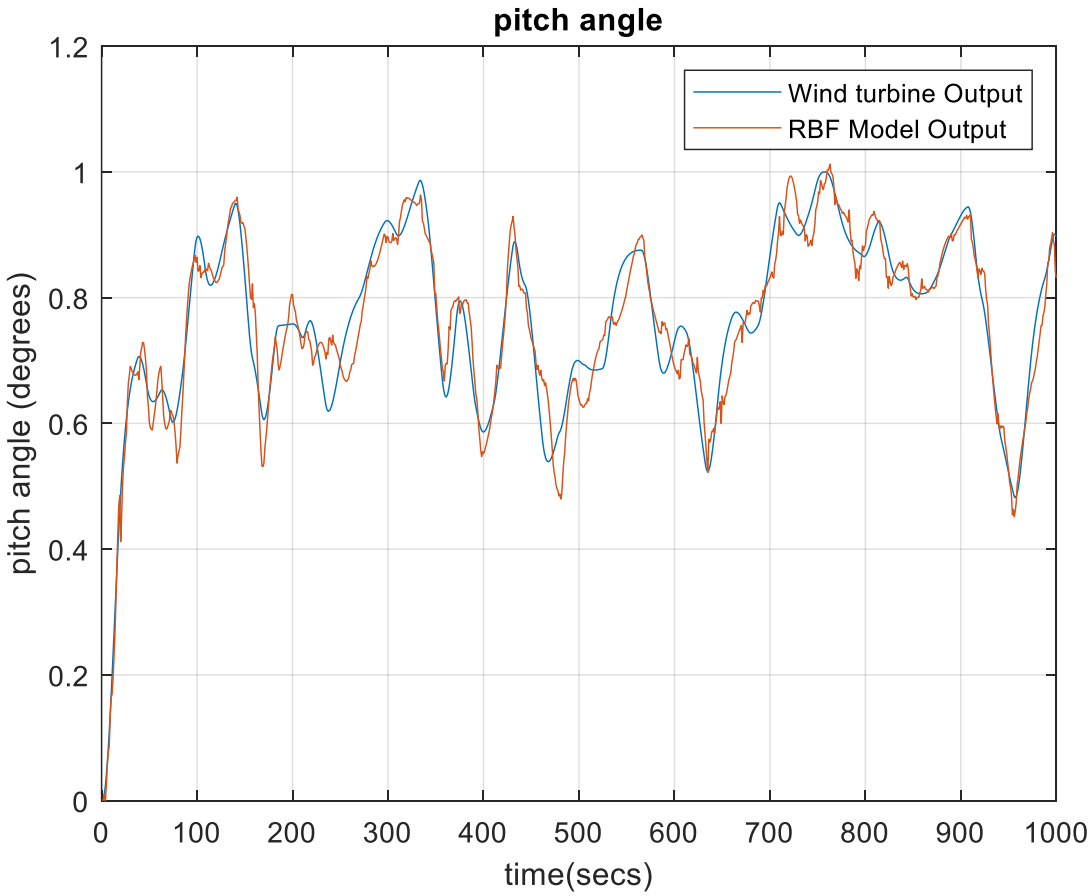

Figure 5.08: Pitch angle training data output.

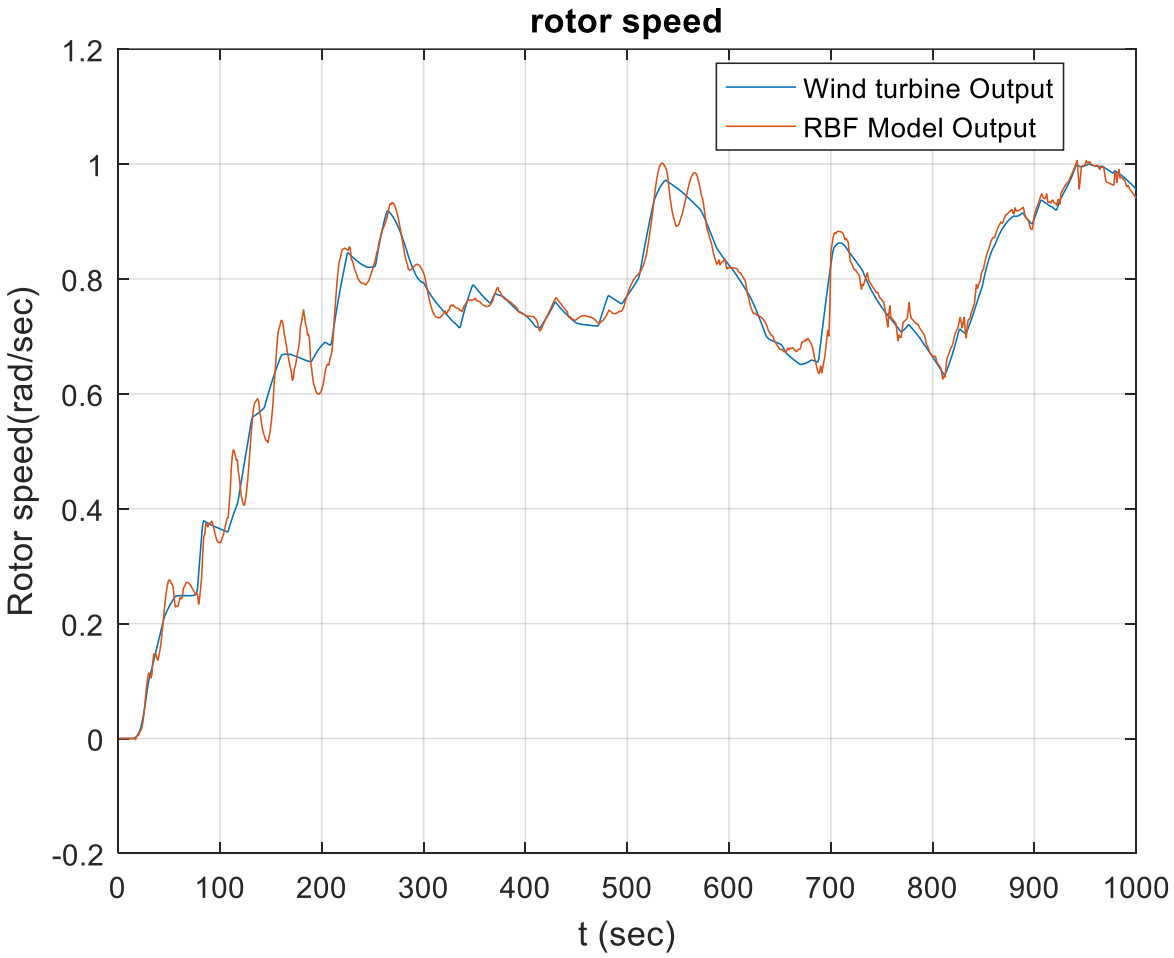

Figure 5.09: Rotor speed training data output

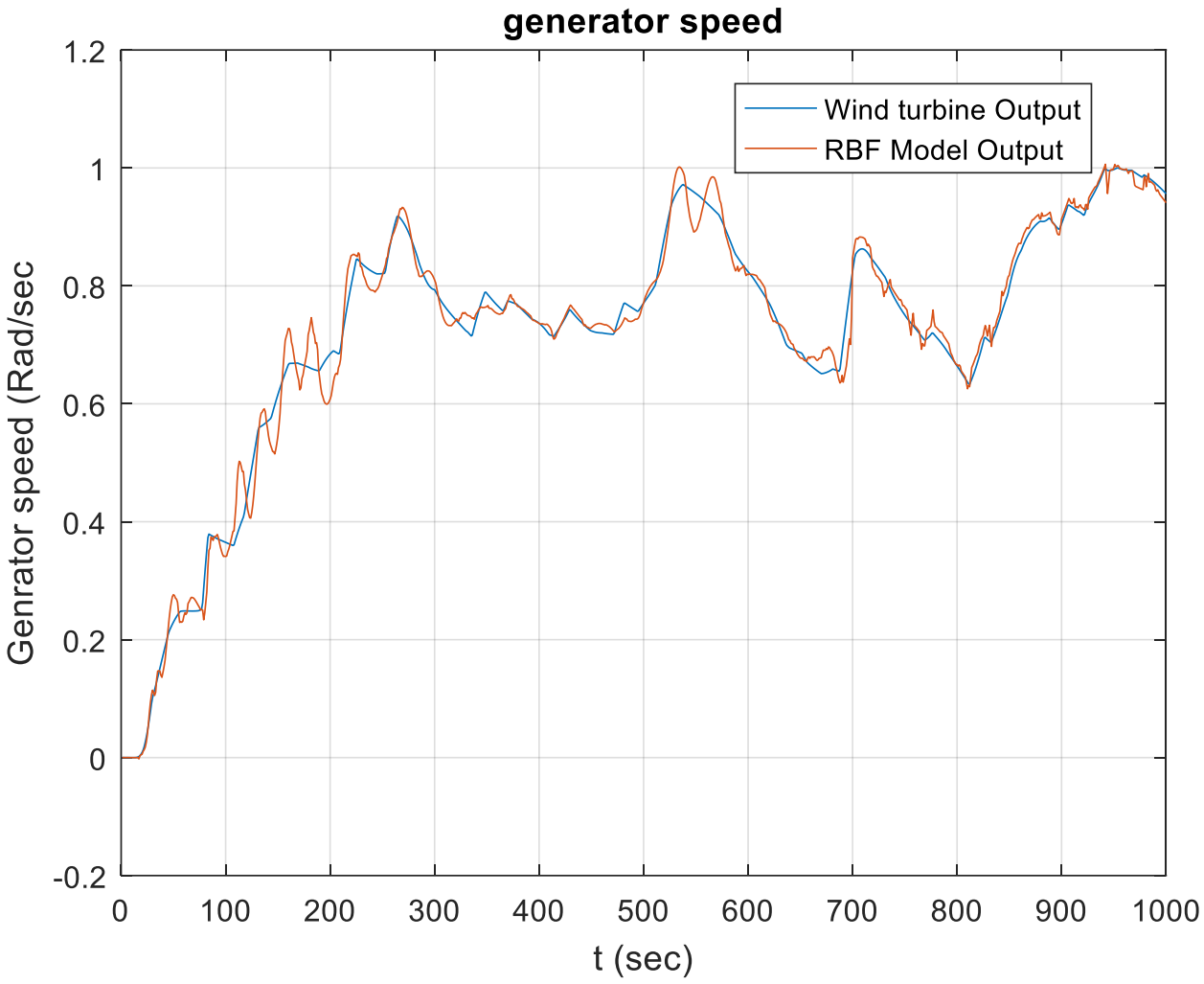

Figure 5.10: Generator speed training data output

5.5 Fault detection using RBF NN

5.5.1 No fault condition

An independent NN model is trained with data collected from the wind turbine at healthy conditions. The modelling error between the wind turbine output and the model prediction will be used as the residual signal. Here the residual is just modelling error caused by noise and model-plant mismatch. This is shown in figure 5.11.

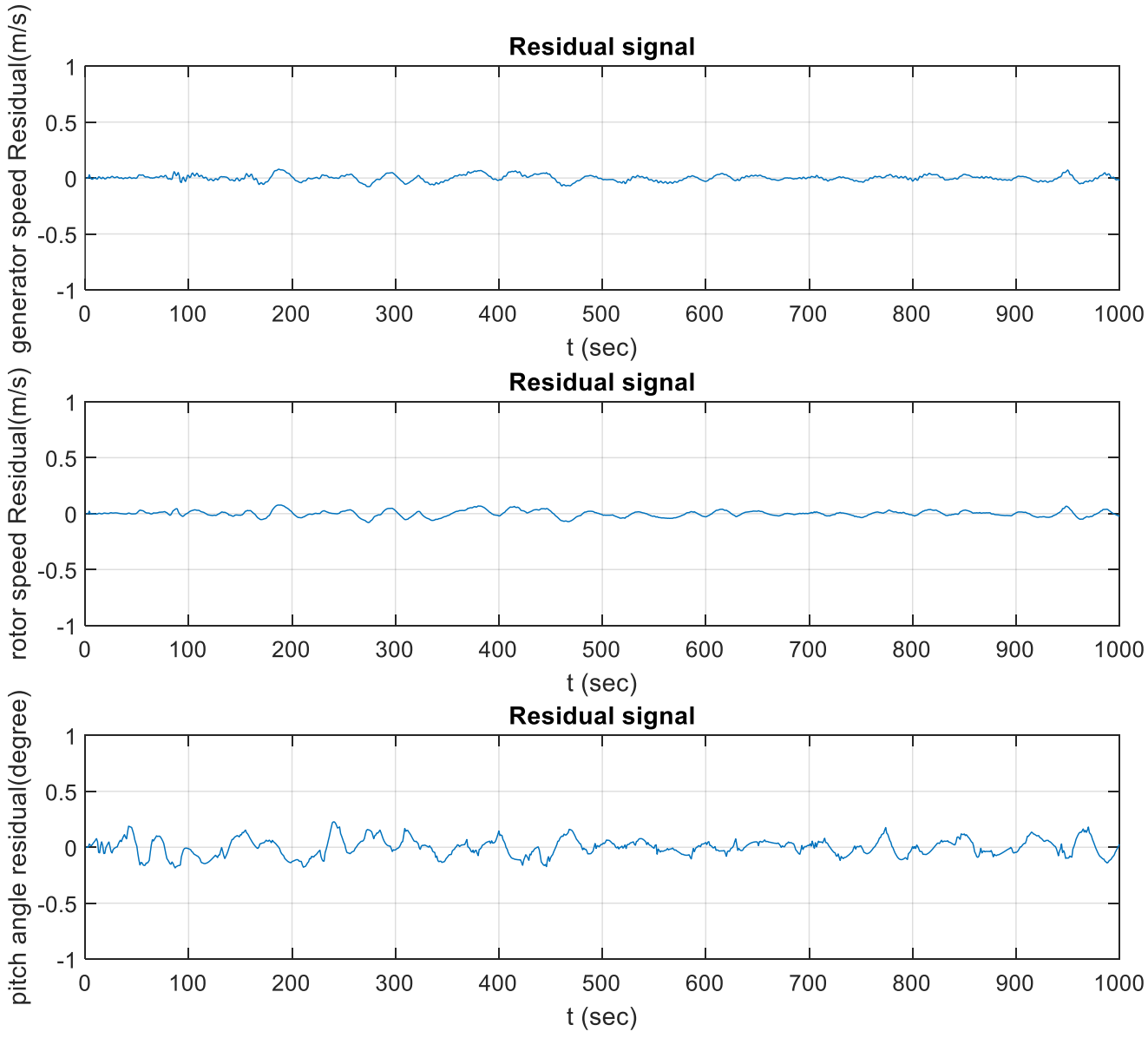

Figure 5.11: Residual signals of the wind turbine

5.5.2 At faulty condition

A total of 5000 data sets were used in testing. Three sensor faults are considered here. They are 20 to 50 percent changes on the output of the wind turbine blade angle, rotor speed and generator speed sensors. The faults occurred as can be presented in Table 5.3. The faulty data for the sensors are generated using multiplying factors of 0.5, 1.5,1 respectively. These faults are simulated with the wind turbine model. Figure 5.12 shows the residual signal of the wind turbine. It can be seen clearly that the three sensor faults could be detected by the designed RBF Neural network.

Table 5.3: Sensor Output fault time

Outputs	Time of fault
Pitch angle	1000-1500
Rotor speed	4000-4500
Generator speed	2500-3000

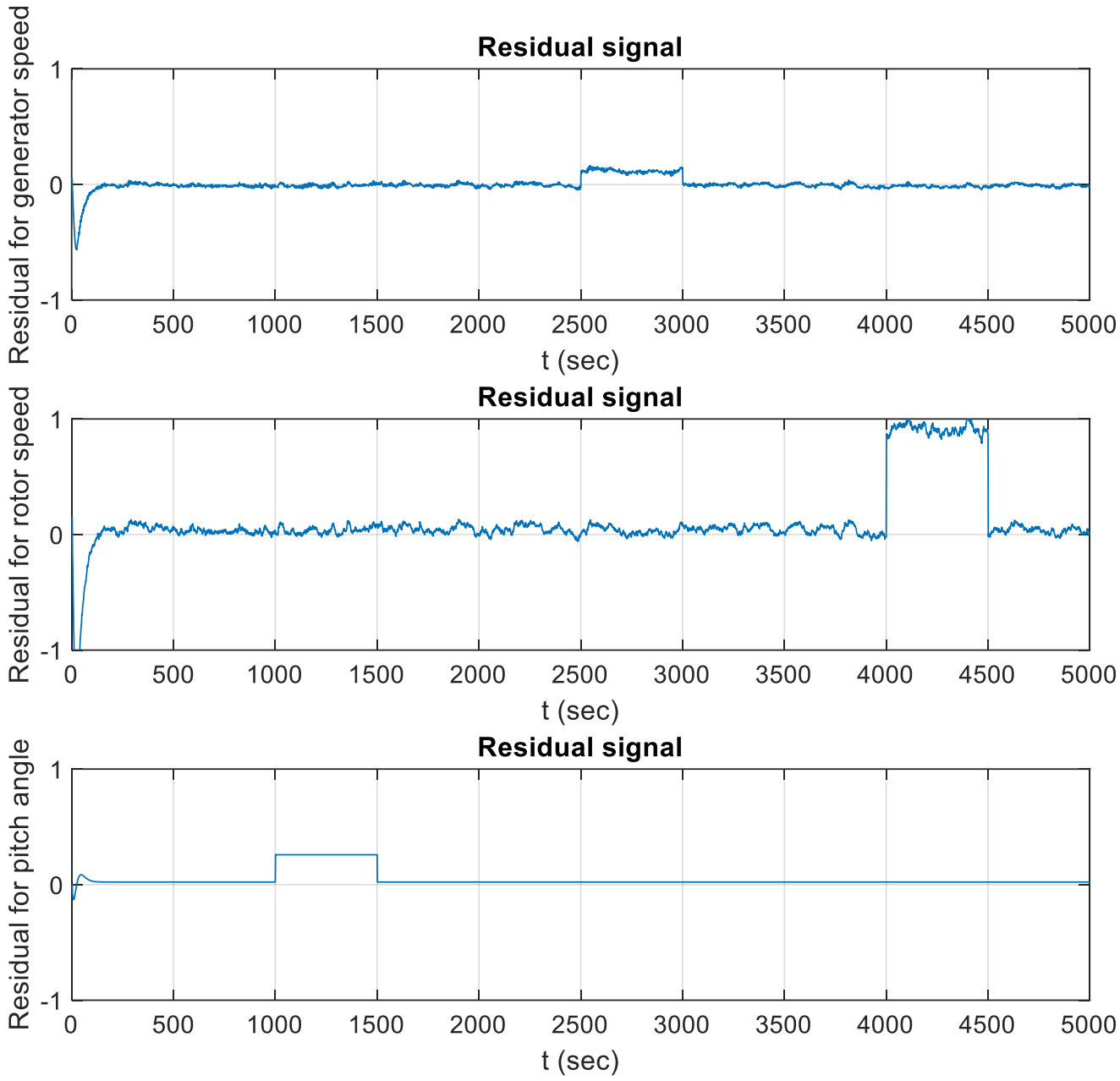

Figure 5.12: Residual signals of a faulty wind turbine

5.6 Fault Isolation using Rbf NN

Here an additional RBF neural network as a classifier is used to isolate all kinds of faults. The classifier has three inputs and four outputs with one representing “No fault”. The inputs of the classifier are the modelling error vector which includes information of all faults. Four sets of data are collected with the first set rid of faults and the other three sets each with one fault only. The training

of the classifier starts with data collection from the wind turbine for each of the faults occurring only; these data are fed to the classifier and the target of the output being “1” for the dedicated output and every other output being “0”. 4000 data samples were collected with 1000 data samples without fault and each set of 1000 with one of the other faults. The centers and widths are chosen using the K-means clustering algorithm and the P-nearest centers method respectively. The number of hidden layers was tried using trial and error and it was chosen to be “6”. Figures 5.13 to 5.15 shows the filtered output of the fault classifier. From these figures it can be observed that the faults were clearly isolated.

Table 5.4: Sample data and fault type

Data Samples	Fault Type
0-999	No Fault
1000-1200	Generator speed sensor fault
1201-1999	No Fault
2000-2200	Rotor speed sensor fault
2201-2999	No fault
3000-3200	Pitch angle sensor fault
3201-4000	No Fault

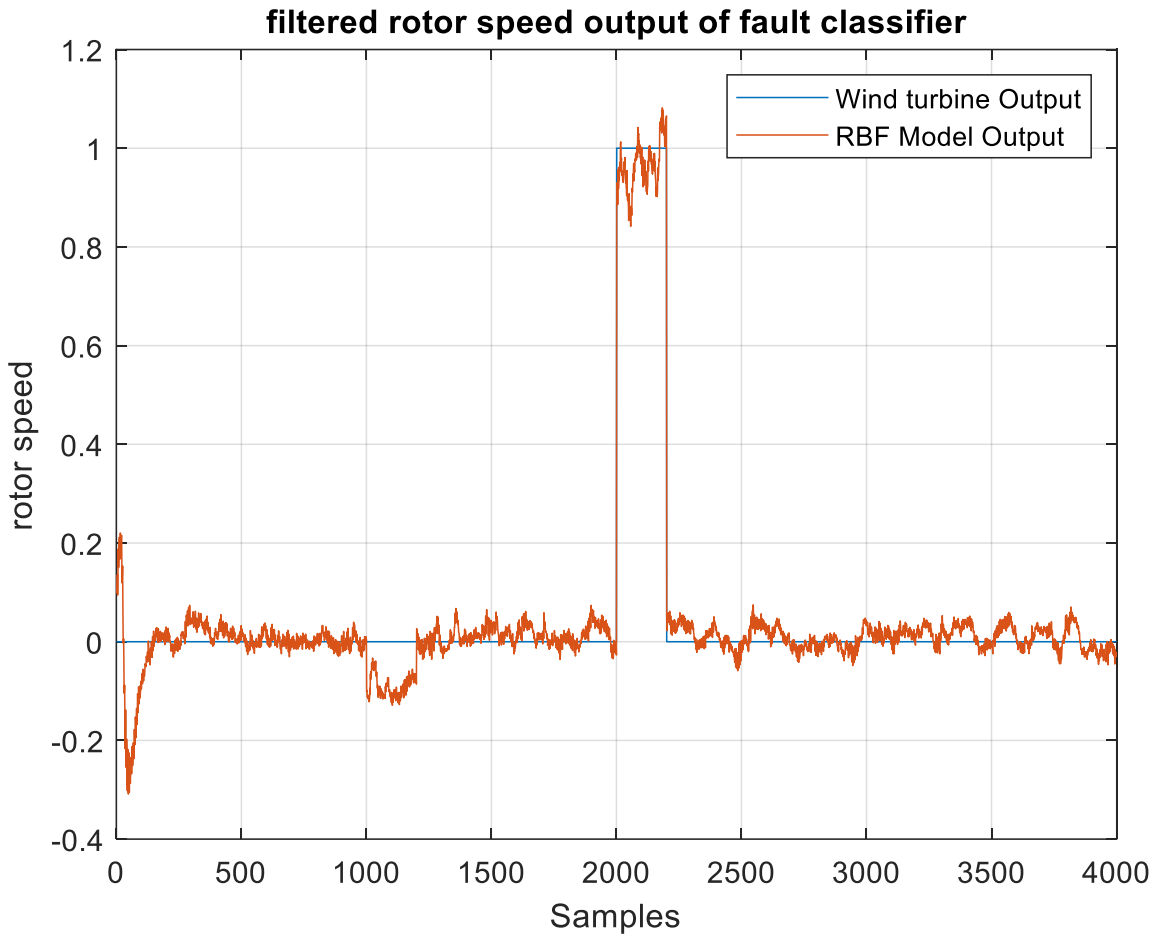

Figure 5.13: Filtered rotor speed output of fault classifier.

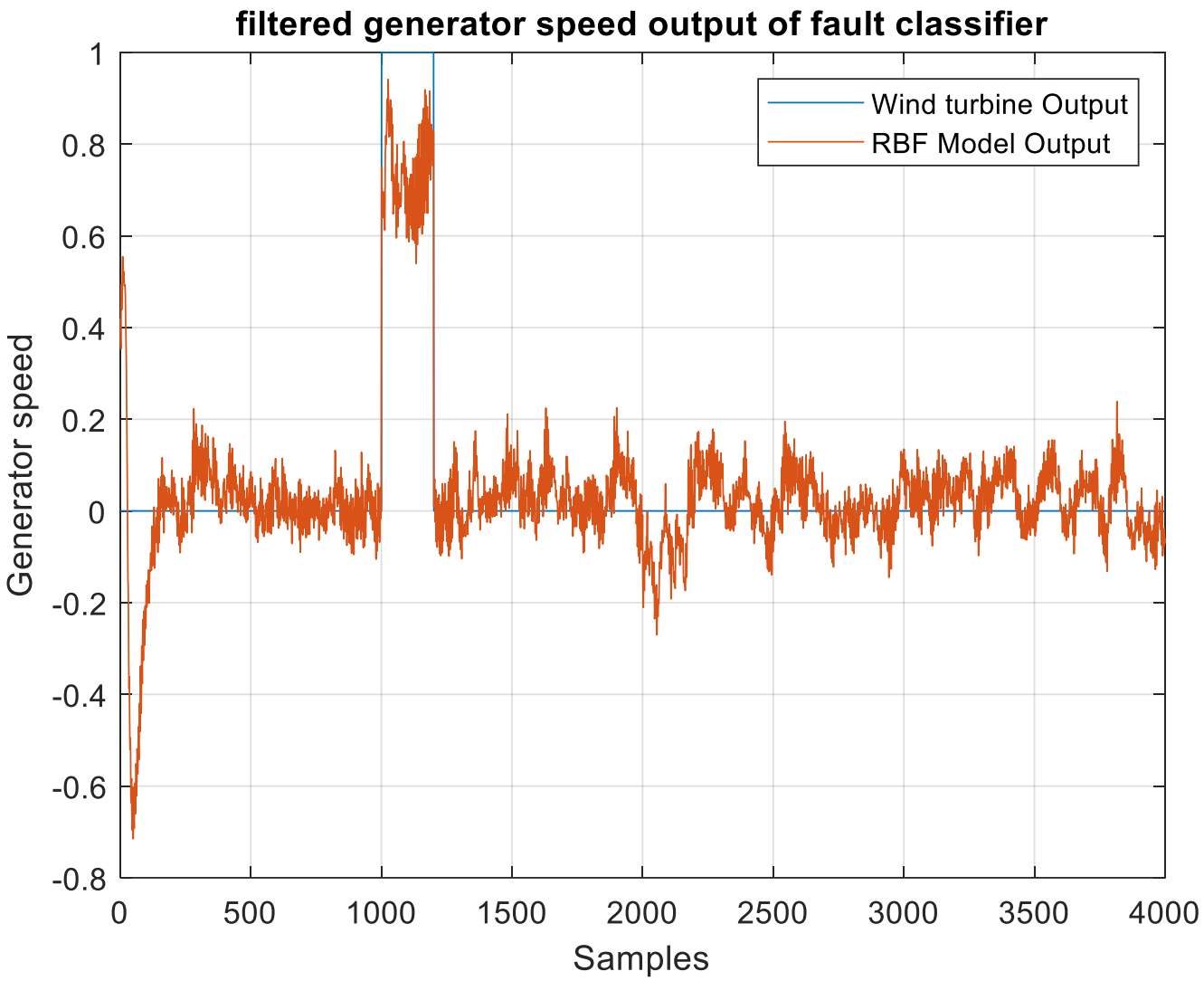

Figure 5.14: Filtered generator speed output of fault classifier.

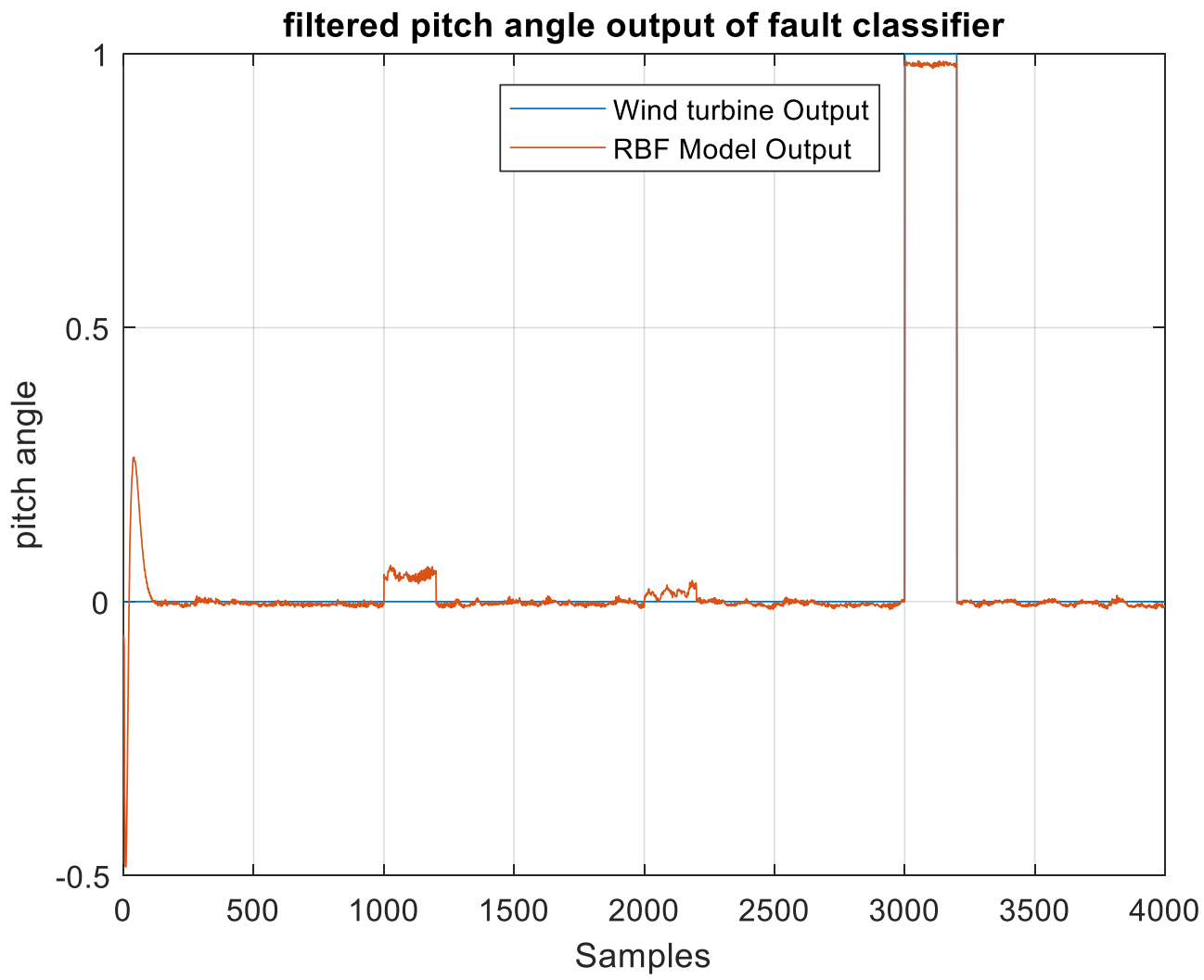

Figure 5.15: Filtered pitch angle output of fault classifier.

5.7 Summary

In this chapter, a robust fault diagnosis approach using radial basis function neural network is proposed. The choice of the RBF is because of its ability to approximate a nonlinear input into a linear output. The training process is also faster and better when compared with other types of NN. The RBF is trained using sample data during a fault free operating condition. The developed method is applied to the benchmark model of the wind turbine system with sensor faults simulated. Simulation results demonstrate that the proposed method is effective with residual signal sensitive to the sensor faults. For the fault isolation an additional RBF NN was used as a classifier. From the simulation results the proposed method successfully isolated all three sensor faults.

CHAPTER 6

Fault Detection and Fault Tolerant Control Scheme

6.1 Introduction

To achieve fault detection and fault tolerant control, a FOTSMC scheme is developed here for a hydraulic pitch angle system and combining the control scheme with a disturbance observer. The disturbance observer will be used for fault detection purposes and to help in the fault tolerant control. To estimate the fault rate of change an RBF network estimator is proposed. This method was applied to a variable speed wind turbine pitch angle system.

6.2 Disturbance observer (DO) for fault detection

The Unknown input observer was used to detect fault in a linear model of the wind turbine system. However hydraulic pitch system is a non-linear system which has influenced the choice of a new fault detection method disturbance observer.

A disturbance observer (DO) is designed to detect faults in the variable speed pitch angle system. A model following the vector -matrix form is presented for system models (3.31).

$$\dot{x_1} = f(x) + g_1(x)u(t) + g_2 d(t) \tag{6.01}$$

Where $x = [0\ 0\ b(x)]$, $g_2 = [0\ 0\ 1]^T$. x is the system state. $u \in R$ is system input, $g(x) and\ b(x)$ are smooth nonlinear function with $g(0) = 0$ and $b(x) \neq 0$. $d(t)$ is the system fault. It is assumed that $\dot{d(t)}$ is bounded and continuous and the knowledge of the bounds is not needed. This assumption is generally satisfied in engineering problems.

A disturbance observer for (6.01) is designed in (6.02)

$$\begin{cases} \dot{z} = -Pg_2 z - P[g_2 Px + f(x) + g_1(x)u] \\ \hat{f} = z + Px \end{cases} \tag{6.02}$$

where $z \in R,\ P \in R^{1\times 3}$ and $\hat{f}$ are the internal state variable, observer gain matrix and output of the observer or the fault estimate respectively. The disturbance observer designed at (6.02) is presented on a Simulink model as can be seen in Figure 6.01.

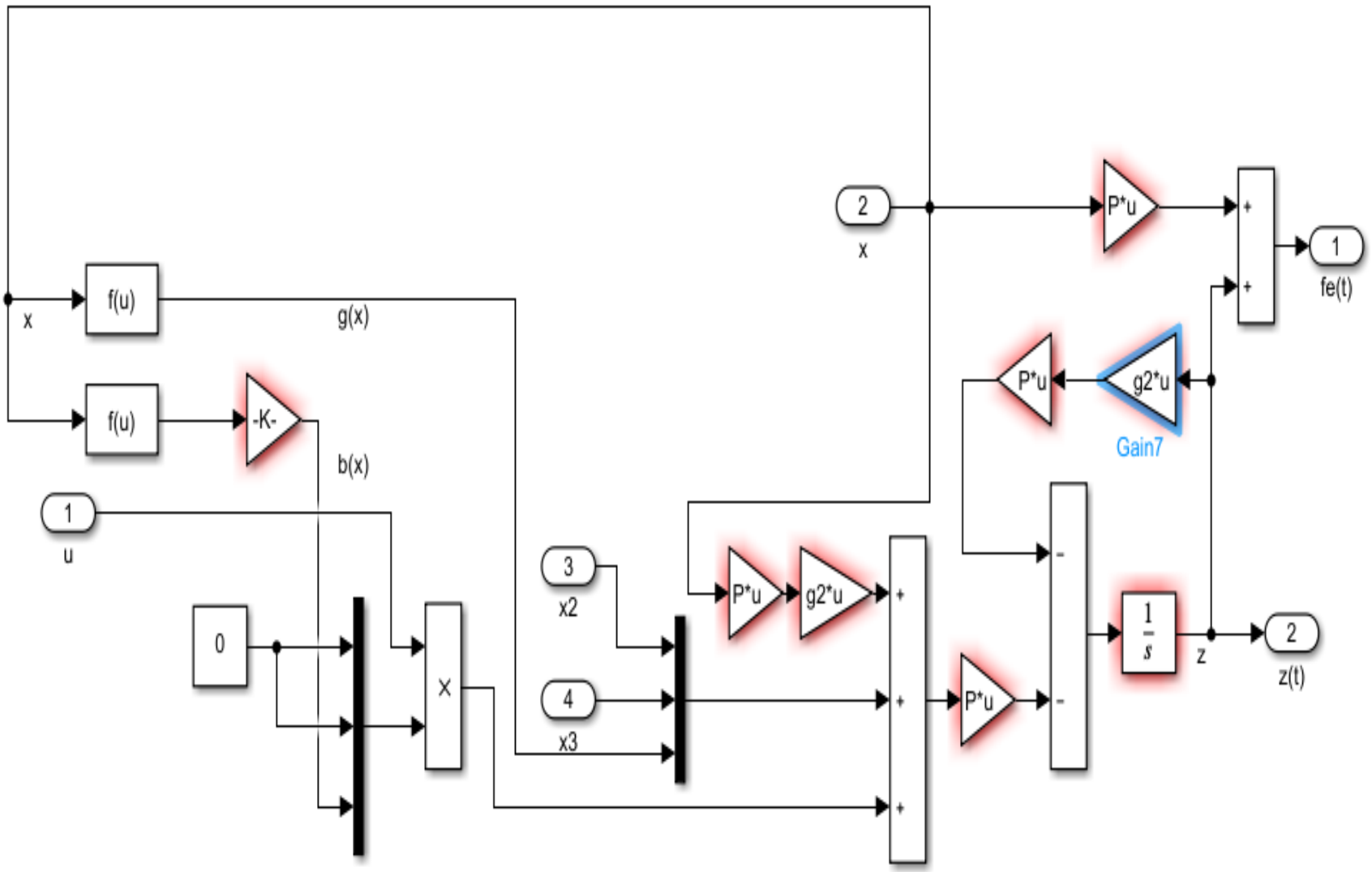

Figure 6.01: Disturbance observer Simulink model.

For the stability and performance of the DO, the fault estimation error defined as (6.03) has the dynamics (6.04)

$$e_f = d - \hat{f} \tag{6.03}$$

$$\begin{aligned} \dot{e_f} &= \dot{d} - \dot{\hat{f}} \\ &= \dot{d} - \dot{z} - P\dot{x} \\ &= \dot{d} - Pg_2 z + Pg_2 Px + Pf(x) + Pg_1(x)u - Pf(x) - Pg_1(x)u - Pg_2 d \\ &= \dot{d} - Pg_2 z + Pg_2 Px - Pg_2 d \\ &= \dot{d} - Pg_2 e_f \end{aligned} \tag{6.04}$$

For (6.04) to be stable, P is designed such that $Pg_2 > 0$.

The fault estimation error will converge to zero if the fault is a constant signal. This means that at steady state the fault estimation is accurate. However, in industrial applications this may not be the case because some faults are time varying. Real faults are related to estimated faults in (6.05). (6.05) is derived from (6.03).

$$\begin{aligned} \hat{f} &= Pg_2 e_f \\ &= Pg_2 d - Pg_2 \hat{f} \end{aligned} \tag{6.05}$$

It is concluded that $\hat{f}$ is a low pass first order filter output of the original fault signal (d). The time constant of the filter is $\tau = Pg_2$. If P is designed small the estimation error can be reduced.

6.3 Fault tolerant control scheme

For the fault tolerant control scheme, a robust full order terminal sliding mode control scheme is designed with an RBF network updated online to compensate the effects of the faults. A disturbance observer is developed to help in fault detection and for fault tolerant control. The control system for the nonlinear hydraulic pitch system is designed in a way that the output x_1 tracks the refrence signal such that when a fault occurs, the stability of the closed loop system is guaranteed with the fault not influencing the tracking performance. The use of FOTSM control technique makes the system

states reach the TSM and then converge to the equilibrium in finite time. The chattering phenomenon is eliminated because the controller uses the forms of integrator or low pass filter.

For an nth order nonlinear system as described in (6.06), a model based FOTSM control model that uses the RBF to approximate the nonlinear function was developed.

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = x_3 \\ \cdots \\ \dot{x_n} = f(x) + d(x) + b(x)u \end{cases} \tag{6.06}$$

Where $x = [x_1, x_2, \ldots, x_n]^T \in R^n$ Is the system state $u \in R$ is the system input, $f(x)$ and $b(x)$ denotes Continuous nonlinear functions with $f(0) = 0$ and $b(x) \neq 0$. The fault signal of the system is denoted as $d(x)$. There is an assumption that the first order derivative of the fault $\dot{d}(x)$ is bounded and a continuous function. The bound of the fault change in the system is unknown.

Definition

A FOTSM manifold definition used here when compared to the definition in [38] looks similar. The difference is that the tracking error is used for the reference tracking control instead of the system state.

$$s = \dot{e_n} + \beta_n e_n^{a^n} + \ldots \beta_1 e_1^{a^1} \tag{6.07}$$

Where e_i $(i = 1,2,\ldots,n)$ as defined below is the tracking error with $r(t)$ being the reference output.

$$\begin{cases} e_1 = x_1 - r \\ e_2 = \dot{e_1} = \dot{x_1} - \dot{r} \\ \quad . \\ \quad . \\ \quad . \\ e_n = \dot{e}_{n-1} \end{cases} \tag{6.08}$$

When β_i $(i = 1,2,\ldots,n)$ is designed the condition that the polynomial $p^n + \beta_n p^{n-1} + \cdots + \beta_2 p^1 + \beta_1$ is Hurwitz is fulfilled. $\alpha_i (i = 1,2,\ldots,n)$ is designed as

$$\begin{cases} a_1 = a_2 & n = 1 \\ a_{i-1} = \frac{a_i a_{(i+1)}}{2a_{i+1} - a_i} & i = 2, \ldots, n \ \forall n \geq 2 \end{cases} \tag{6.09}$$

Where $a_{n+1} = 1,\ a_n = a,\ a \in (1-\varepsilon, 1), \epsilon \in (0,1)$.

Once a FOTSM is achieved s=0 is reached and the tracking error in (6.08) will converge to zero and converge to the terminal sliding mode within finite time. The scheme introduced in [34] is designed to achieve convergence of system state in (6.06) in finite time. The scheme needs previous knowledge of the upper bounds $d(x)$ and $\dot{d}(x)$. This may not be available for some practical applications. A new method will be developed to guarantee finite time convergence of the tracking error and system states without knowledge of fault and fault derivatives. The RBF estimator will be used to approximate the fault derivative. This new method combines the online estimation of the fault change rate by an adaptive RBF network and a model FOTSM to drive the states onto the sliding surface.

The RBF network is used to estimate the fault derivative $\dot{d}(x)$ from the current system state and it is defined as (6.10).

$$\dot{d}(x) = W^T \emptyset(y) + \epsilon \tag{6.10}$$

Where the approximation error is denoted as ϵ , $y = [x_1, \dots, x_n, \dot{x_n}]$ is the RBF input vector. The nonlinear basis function is presented as $\emptyset(y) = [\emptyset_1(y), \emptyset_2(y), \dots, \emptyset_i(y)]^T \in R^l$, $W = [w_1, w_2, \dots, w_i]^T \in R^l$ is the weight vector. The Gaussian function can be expressed in (6.11)

$$\emptyset(y) = \exp\left(-\frac{||y-c_i||^2}{\omega_i^2}\right) i = 1, \dots, l \tag{6.11}$$

Where c_i is the centre of the ith hidden layer node. w_i is the radius of the ith Gaussian function.

The controller is designed below based on the RBF neural network and the FOTSM manifold in (6.07)

$$u = b^{-1}(x)(u_1 + u_2) \tag{6.12}$$

$$u_1 = -f(x) - \beta_n e_n^{a_n} - \dots - \beta_1 e_1^{a_1} + \ddot{r} \tag{6.13}$$

$$\dot{u_2} = -W^T \emptyset(y) - (k+\eta) sgn(s) \tag{6.14}$$

Where $K > \epsilon_N$, $\eta > 0$, the signum function is denoted as sgn(0). Using the information provided by the states in the network input vector $z(x)$, the RBF estimator is adapted online. The adaptation law is designed in (6.15)

$$\dot{\widehat{W}} = \frac{1}{\gamma} s\emptyset(y) \tag{6.15}$$

Where $\gamma > 0$ is the inverse of a learning rate. The post fault stability of the closed-loop system given the designed control law and the adaptation law of the Rbf network is stated by the theorems below.

Theorem 1

In (6.06), If the Fotsm manifold is chosen as (6.07) and the control law and network adaptation law are designed as in (6.12) – (6.15) then the state of the closed loop system will converge to zero which makes the system stable.

7.3.1 Lyapunov Stability test

The following Lyapunov function will be used

$$V = \frac{1}{2}s^2 + \frac{1}{2}\gamma\widetilde{W}^T\widetilde{W}$$

Where $\widetilde{W} = W^T - \widehat{W}$ is the RBF network weight convergence error.

$$\dot{V} = s\dot{s} + \gamma\widetilde{W}^T\widetilde{W} \tag{6.16}$$

Substitute (6.06) into (6.07)

$$s = f(x) + d(x) - \ddot{r} + b(x)u + \beta_n e_n^{a_n} + \cdots + \beta_1 e_1^{a_1}$$

From (6.12) and (6.13)

$$s = f(x) + d(x) - \ddot{r} + u_1 + u_2 + \beta_n e_n^{a_n} + \ldots + \beta_1 e_1^{a_1}$$

$$s = f(x) + d(x) - \ddot{r} - f(x) - \beta_n e_n^{a_n} - \ldots - \beta_1 e_1^{a_1} + \ddot{r} + u_2 + \beta_n e_n^{a_n} + \ldots + \beta_1 e_1^{a_1}$$

$$= d(x) + u_2 \tag{6.17}$$

The time derivative of (6.17) gives

$$\dot{s} = d(\dot{x}) + \dot{u_2} \tag{6.18}$$

Substituting (6.10) and (6.14) in (6.18) gives us

$$\dot{s} = W^T \emptyset(y) + \epsilon - \widehat{W}^T \emptyset(y) - (k + \eta) sgn(s)$$

$$\dot{s} = \widetilde{W}^T \emptyset(y) + \epsilon - (k + \eta) sgn(s) \tag{6.19}$$

Substitute (6.19) and (6.15) into (6.16)

$$\dot{V} = s\widetilde{W}^T \emptyset(y) + s\epsilon - s(k + \eta) sgn(s) + \gamma \widetilde{W}^T \widetilde{W}$$

$$= \widetilde{W}^T [s\emptyset(y) - \gamma \dot{\widehat{W}}] + s\epsilon - s(k + \eta)|s|$$

$$= s\epsilon - (k + \eta)\,|s|$$

$$\leq (\epsilon - k - \eta)|s| \tag{6.20}$$

k and η are constants to be selected for the design. For $\dot{V} < 0$, k and η are chosen such that $\eta > 0, k > \epsilon_N \geq \epsilon, then\ (e - k - \eta)|s| < 0\ for\ s \neq 0$. V will converge to zero according to the Lyapunov stability theorem. Following the definition of V in (6.16) and the theorem of terminal sliding mode, the system states will converge to zero on the sliding manifold in finite time.

The RBF estimator weights converge to the existing optimal weights. (6.14) adopts the form of an integrator. The sign function was used in $\dot{u_2}$ instead of the u_2 to ensure that the discontinuity caused by the sign function will not directly affect the control law but will be integrated. The control signal will be guaranteed continuous. The chattering problem is eliminated due to this technique. The derivative of s is unused causing the designed control to avoid singularity. The information generated in fault detection is used to assist for fault tolerant control.

6.4 Fault tolerant control of a hydraulic pitch angle control

The pitch angle control developed here with fault detection and fault tolerance is implemented using computer simulation. Here we used a step reference pitch angle to represent abrupt change of the wind speed. The abrupt fault to be simulated is a step change of friction torque. The simulation was carried out in matlab/Simulink. The pitch angle plant and the disturbance observer are designed in Simulink while in matlab file(m-file) the RBF network and the control algorithm are designed. The matlab file calculates the control input, calls the Simulink model to obtain the plant output and the

fault observer output in each sample period. The sample period used in this research is 0.001s and the simulation ran for 8secs.

Using (3.31) to (3.34) a Simulink model is designed for a servo hydraulic drive plant and the disturbance observer is designed using (6.01) to (6.05). The Simulink model has two inputs and five outputs. The inputs are the friction change and control input while the outputs are the pitch angle system output, the other two states of the motor shaft rotary speed and rotary acceleration, fault to be detected and the estimated fault. The physical parameters in Table 3.2 were used.

A mat lab file was created to calculate the sliding surface according to (6.07) to (6.09), control variable according to (6.12) to (6.14) and RBF neural estimator output according to (6.10) and (6.11). The weight is updated according to (6.15). The faults are simulated by changing the viscous frictional torque on motor axis, $T_P(t)$ as below to mimic the change in friction.

The sliding surface is designed here n=3, so the sliding manifold in (6.07) is generated with the following parameters as stated in Table 6.1. To make sure $p^3 + \beta_3 p^2 + \beta_2 p + \beta$ is Hurwitz the three poles are set to $P_{1,2,3} = 3.2$

Table 6.1: Design parameters of the sliding surface

Parameters	values
α_1	0.300
α_2	0.3913
α_3	$\frac{9}{16}$
α_4	1
β_1	32.7680
β_2	30.7200
β_3	9.600

The RBF network with the centres is designed as follows;

$$c = [c_1, c_2, c_3, \ldots c_{10}] = \begin{bmatrix} 0.1 & 0.2 & 0.3 & \cdots & 1.0 \\ 0.05 & 0.15 & 0.25 & \cdots & 0.95 \\ 0 & 0.1 & 0.2 & \ldots & 0.9 \\ 0.02 & 0.12 & 0.22 & \ldots & 0.92 \end{bmatrix}$$

$$w = [w_1, w_2, \ldots w_{10}] = [0.5, 0.5, 0.5 \ldots 0.5]$$

Initial value for the weight is $W = [1,1, \ldots,1]^T * 10^{-5}$

The control u(t) is designed using (6.12) where u_1 is from (6.13) and u_2 is from the integration of (6.14). The control parameters in (6.20) is chosen as k=200,$\eta = 90$ and the learning rate γ is chosen as 1. The Rbf estimator is designed using (6.15). The fault signal used is a square wave to mimic abrupt fault. The simulation results are presented in Figure 6.02 and Figure 6.03.

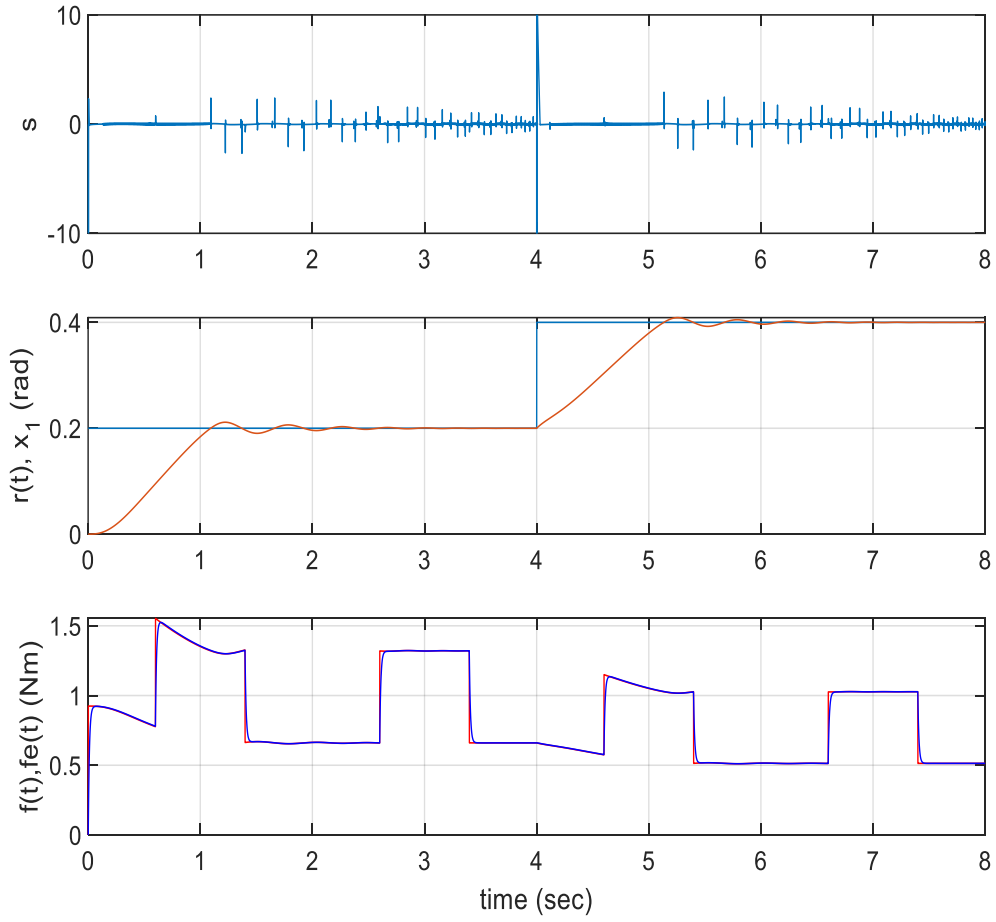

Figure 6.02: Pitch system control performance using step reference and fault

There are three Figures in Figure 6.02. The first Figure shows the sliding mode ‘s’ while the middle figure shows the controlled pitch angle and the pitch angle reference signal. These two signals are plotted together for comparison. The last figure presents the output of the disturbance observer which is the actual fault and estimated fault. Figure 6.02 clearly shows that the sliding mode approaches zero from the non-zero initial value. During fault, the sliding mode moves away from zero slightly but still maintains a zero value throughout the simulation time. In the second figure, the pitch angle which is the first state converges to the reference output even with the occurrence of fault as we can see in the last Figure. This clearly shows that the fault tolerant control was achieved successfully. The last Figure in Figure 6.02 shows that the estimated fault follows the real fault very closely when it occurred in the system. This clearly shows that the estimated faults can be used to report fault size and time of occurrence.

In Figure 6.03, the rotary speed on hydraulic motor axis and the rotary acceleration which is the second and third state respectively are displayed in the first and second Figures. The two states are bounded on the effect of the RBF network estimator. This is guaranteed because of the Lyapunov theory that was used when deriving the control law and the network weight updating law. The last Figure displays the control variable where there is no chattering. Even though the specific fault has been tolerant in the pitch angle performance. it is important to use the fault diagnosis function to notify an operator to plan a detailed maintenance and inspection.

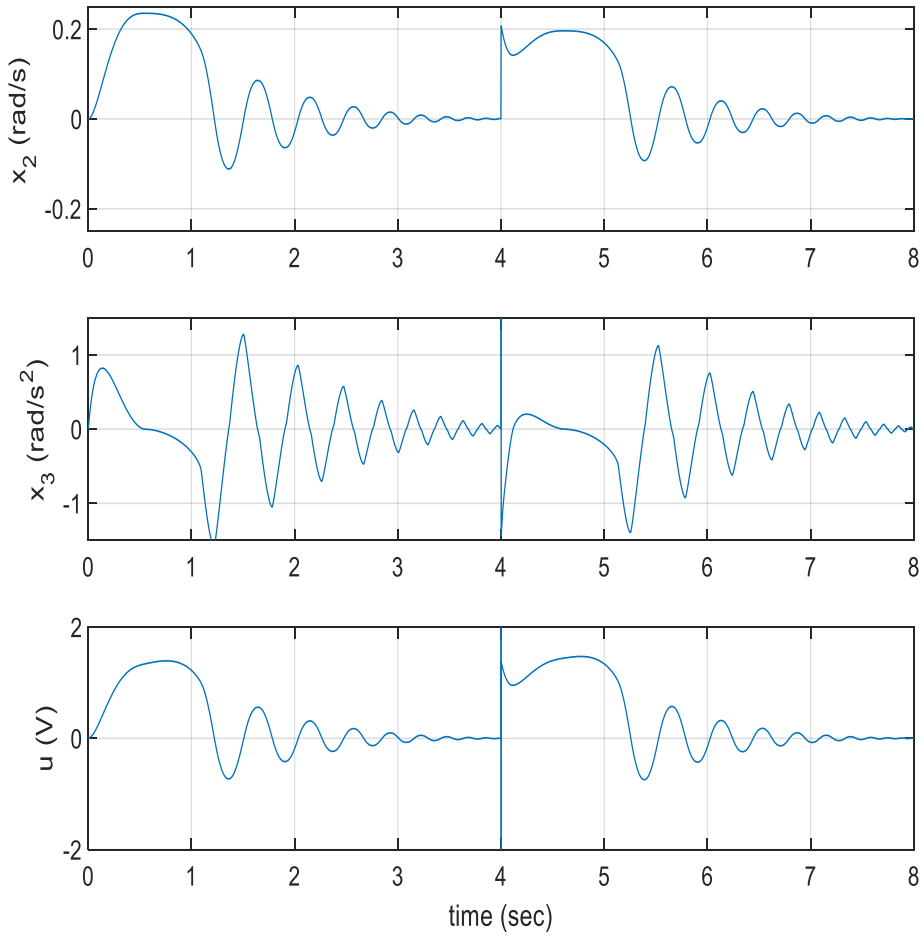

Figure 6.03: Pitch angle control performance with step reference and fault

6.5 Summary

A fault tolerant control system has been designed in this work using the FOTSMC and adaptive RBF estimator. A disturbance observer was designed for the purpose of reporting fault occurrence, estimate fault size and time of occurrence. The developed fault tolerant system uses the Lyapunov method to guarantee convergence and stability. This developed fault tolerant system was applied to a variable speed wind turbine generator to control the pitch angle in the presence of faults. The fault signal used is a square wave to mimic abrupt fault.

CHAPTER 7

Conclusion and Further works

7.1 Conclusion

In this research, a FD and FTC of the pitch system of a wind turbine using full order terminal sliding mode control is studied. This research is divided into three phases.

In the first Phase, the main aim is to design a robust fault detection approach using unknown input observer. A benchmark wind turbine system is modelled. This system consists of three subsystems: pitch actuator, aerodynamics and drive train system. The aerodynamics converts the wind kinetic power into mechanical power. The angle of the wind turbine blade is controlled by the pitch to extract the most power from the wind. The drive train is used to transfer aerodynamic torque from the rotor to the generator. The drive train model is modelled as a rotational 2-mass, 1-spring.1-damper system. The drive train of a wind turbine system consists of a main gear box, low speed shaft (turbine) and a high-speed shaft (generator). The pitch dynamics and the drive train dynamics were combined to derive the linear model of the wind turbine benchmark model. These models were simulated on matlab/Simulink. A robust observer has been designed. This observer known as an unknown input observer is sensitive to faults in the state equation while robust to system disturbances. Faults were simulated into the system after the unknown input observer was designed. These faults include pitch system actuator fault and three sensor faults which are rotor speed sensor, generator speed sensor and pitch angle sensor. A disturbance was also introduced to the system. For the actuator fault, it is assumed that this fault is caused by the malfunction of some component and the value of the generator torque increased by 70 percent. The consequence of this fault is that the torque control is slow. The three sensor faults are also simulated. The pitch angle measure is the first sensor fault which is assumed to occur between 20- 30 secs while the generator speed which is the second fault is assumed to have a 50 percent drop on the measurement during the time 40 to 50 secs. The last sensor fault is assumed to occur between 30 and 40 secs. From the simulation results, when the system operates at no fault condition the residuals and state error remain zero. When the faults are simulated the fault detection residual was robust to disturbances and sensitive to the simulated faults. The simulation models and the faults were set up using Simulink/Matlab. The observer was developed based on sets of equations on Simulink. This phase was done to verify if the fault in the linear model of the wind turbine can be simulated.

In the second phase of this thesis, a fault diagnosis scheme was designed for a wind turbine system using an independent RBFNN. The choice of the RBF was made because of its ability to approximate a nonlinear input into a linear output. The training process is also faster and better when compared with other neural networks. Three sensor faults are simulated in the wind turbine model. In this phase, two different techniques employed RBFNN to investigate the fault diagnosis. The first technique is a using an independent RBFNN for generating residuals. The generated residuals were used in detecting the three sensor faults. A set of random amplitude signals of wind speed, reference blade pitch angle and generator torque are generated. This is generated randomly to cover the whole range of frequencies and entire operation space of amplitude in the wind turbine. All the raw data samples are scaled into the range [0, 1] to increase the accuracy of the RBF and decrease the error. The center is calculated using the k-means algorithm, the weight was chosen using the p nearest neighbour method and the hidden layer nodes of the RBFNN was chosen to be 8. The recursive least square is used here to determine the RBF network weights W. An independent NN model is trained with data collected from the wind turbine at healthy conditions. The model is used parallel to the wind turbine in online mode to be able to predict wind turbine output. The RBFNN model designed was trained and tested on matlab/Simulink and the result shows a good match between the wind turbine output and the RBF model output. The developed method is applied to the benchmark model of the wind turbine system with sensor faults simulated. Simulation results demonstrate that the proposed method is effective with residual signal sensitive to the sensor faults. The second technique is applying an additional RBFNN as a classifier for fault isolation purposes. The simulation result shows that the three sensor faults were successfully classified. The training of the RBFNN was carried out on matlab and Simulink.

The third phase of this research is focused on the fault diagnosis and fault tolerant control of the pitch system of the wind turbine using full order terminal sliding mode control. For the fault tolerant control, a full order terminal sliding mode control and adaptive RBF estimator are applied to a pitch angle control of wind turbine system against the system fault of friction torque change. For the fault diagnosis, a disturbance observer is designed to report the occurrence of fault and to estimate the fault size and time. The full order terminal sliding mode control when combined with RBF estimator reduces the limitation that requires the known bound of fault change rate. The Lyapunov method was used to guarantee system stability and the network convergence under post-fault dynamics. A new electrohydraulic servo pitch system of a wind turbine is modelled. This developed fault tolerant system was applied to a variable speed wind turbine generator to control

the pitch angle in the presence of fault. The fault signal used is a square wave to mimic an abrupt fault. The effectiveness of the developed system is evaluated on Matlab/Simulink. The results from the simulation show the pitch angle tracks the square reference and tolerates the fault occurrence.

7.2 Further works

The further works of this thesis are outlined below,

i. The faults in a wind turbine system are more than the three sensor faults studied in this thesis for the fault diagnosis of the wind turbine using RBF neural network. As a further study more faults could be simulated.
ii. To test the RBF neural network with a real test rig because fault test on simulation is less complicated than fault test on real application.
iii. To set up a hydraulic pitch system real test rig with all sensors and actuators present.
iv. For the developed fault tolerant control method, the reliability and performance can be tested on a real time application.

REFERENCES

[1] Liu Y, Yu D (2014), “Robust fault detection for wind turbine systems” Proceedings of the 20th international conference on Automation and computing, Cranfield university, Bedfordshire, UK. 12-13 September.

[2] Martinez J (2007), “Modelling and control of wind turbines” Department of chemical engineering and chemical technology Imperial college London.

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

[4] Polycarpou M (2000), “Fault Accommodation of a Class of Multivariable Nonlinear Dynamical Systems Using a Learning Approach” IEEE transactions on automatic control, vol. 46, no. 5,

[5] Van, M., Shuzhi, S., and Hongliang, R. (2017), “Robust Fault-Tolerant Control for a Class of Second-Order Nonlinear Systems Using an Adaptive Third-Order Sliding Mode Control”, IEEE trans. on systems, man and cybernetics: systems, vol. 47, no. 2,.

[6] Xu, S., Chen, C., and Wu, Z. (2015)”, Study of Non-singular Fast Terminal Sliding-Mode Fault-Tolerant Control”, IEEE Trans. On Industrial Electronics, Vol. 62, No. 6,.

[7] P. Shi, B. Jiang, and K. Zhang, Cocquempot V (2013),” Observer-Based Fault Estimation and Accommodation for Dynamic Systems” Berlin, Germany: Springer- Verlag, 2012.

[8] Pourmohammad, S. and Fekih, A. (2011), “Fault-Tolerant Control of Wind Turbine Systems-A Review, Proc. of Green Technologies Conference” (IEEE-Green), 2011 IEEE [Online] Baton Rouge, LA, USA.

[9] Zhang, X., Polycarpou, M., and Parisini, T. (2002), “A Robust Detection and Isolation Scheme for Abrupt and Incipient Faults in Nonlinear Systems”, IEEE trans. on automatic control, Vol. 47, no. 4,.

[10] Hamayun, T., Christopher, E. and Halim, A. (2014), Augmentation Scheme for Fault-Tolerant Control Using Integral Sliding Modes, IEEE Trans. on Control Systems Technology, Vol. 22, No. 1, pp. 307-313.

[11] Ding S (2008), “Model-based fault diagnosis techniques: design schemes, algorithms, and tools” Springer-Verlag Berlin Heidelberg.

[12] Gao, z., Cecati, C., and Ding, S.,(2015), “A survey of fault diagnosis and fault-tolerant techniques-part1: Diagnosis with model-based and signal-based approaches” IEEE transactions on industrial electronics, vol 62, No. 6,pp 3757-3767.

[13] Shanbin, L., Dominique, S., Christophe, A. and Joseph, Y. (2008), Stability Guaranteed Active Fault-Tolerant Control of Networked Control Systems, Journal of Control Science and Engineering, Vol.2008, Article ID 189064, 9 pages doi:10.1155/2008/189064.

[14] Van, M., Franciosa, P., and Ceglarek, D. (2016), Fault Diagnosis and Fault-Tolerant Control of Uncertain Robot Manipulators Using High-Order Sliding Mode, Mathematical Problems in Engineering, Vol. 2016, Article ID 7926280, 14 pages.

[15] Yong F, Xingho yu, and Fenging H (2013), ”On nonsingular terminal sliding-mode control of nonlinear systems” Automatica 49,1715-1722.

[16] Spurgeon, Sarah K. (2014), “Sliding mode control: a tutorial,” European Control Conference (ECC 2014), Strasbourg, France.

[17] Zavala, E, Moreno J (2017), “Homogeneous High Order Sliding Mode design: A Lyapunov approach” Automatica 232-238.

[18] Kaur, S., Janardhanan, S. (2014), “Second order sliding mode controller and online path planning for space rovers”, International Journal of Modelling, Identification and Control, Vol. 21, Issue 3, pp. 321-329.

[19] Edwards C and Yuri B (2016), “Adaptive continuous higher order sliding mode control” Automatica 65,183-190.

[20] Wu D, Gao C, and Zhai Y et al (2016), “Fault diagnosis of pitch sensor bias for wind turbine based on the multi-innovation kalman filter” Proceedings of the 35th Chinese Control Conference Chengdu, China.

[21] Wu D, Wen L, Song J and Shen Y (2016), “Fault estimation and fault-tolerant control of wind turbines using the SDW_LSI algorithm”.

[22] Vidal, Y., Tutivén, C., Rodellar, J., and Acho, L., (2015), “Fault diagnosis and fault-tolerant control of wind turbines via a discrete time controller with a disturbance compensator” Energies 2015, 8, 4300-4316; doi:10.3390/en8054300

[23] Odgaard, P.F., and Stoustrup, J. (2012), “Fault tolerant control of wind turbines using unknown input observers. In Proceedings of the IFAC symposium on fault detection, supervision, and safety of technical processes (pp.313–318). Mexico City, Mexico.

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

[25] Viveiros, C., Melicio R., Igreja J., and Mendes V., (2014), “Fuzzy, Integer and Fractional-order Control: Application on a Wind Turbine Benchmark Model” 19th International Conference on Methods and Models in Automation and Robotics (MMAR).

[26] Ashwini P, and Aroukna T (2016), “Mathematical Modeling of Wind Energy System Using Two Mass Model Including Generator Losses” International Journal of Emerging Trends in Electrical and Electronics Vol. 12, Issue. 2,.

[27] Wang, S.H., Davison, E.J. and Dorato, P. (1975), “A parameter insensitive technique for aircraft sensor fault analysis.” journal. of Guidance, Control and Dynamics, Vol.10, pp. 359-367.

[28] Utkin V, Guldner J, and Shi J (1999), “Sliding Mode Control in Electro-Mechanical Systems” Taylor and francis ltd, 11 fetter lane, London.

[29] Utkin V, and Shi J (1996), “Integral Sliding Mode in Systems Operating under Uncertainty Conditions” proceedings of the 35th conference on decision and control kobe, Japan.

[30] Isermann R. (2005) “Model-based fault-detection and diagnosis – status and applications.” Annual Reviews in Control, 29, 71-85.

[31] Hamdapoui R, and Guesmi S (2013) “UIO based robust fault detection and estimation” International Conference on Control, Decision and Information Technologies (CoDIT).

[32] ZHAI, Y. and YU, D. (2008) “Radial-basis-function-based feedforward—feedback control for air—fuel ratio of spark ignition engines” Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 222, 415-428.

[33] KAMAL, M. M., YU, D. W. and YU, D. L. (2014) “Fault detection and isolation for PEM fuel cell stack with independent RBF model” Engineering Applications of Artificial Intelligence, 28, 52-63.

[34] Feng Y, Han F and Yu X (2014) “Chattering free full-order sliding mode control” Automatica 2014;50(4): 1310-1314.

[35] Krzysztof P (2008) “Artificial neural networks for the modelling and fault diagnosis of technical processes” Springer-verlag heidelberg, e-ISBN 978-3-540-79872-9.

[36] Weitian C and Mehrdad S (2005) “An Actuator Fault Isolation Strategy for Linear and Nonlinear Systems” American Control Conference June 8-10, 2005. Portland, OR, USA.

[37] Damiano R, Fatiha N and Vicenc P (2018) “Passive and Active FTC Comparison for Polytopic LPV Systems”

[38] Feng Y, Zhou M, Zheng X, Han F and Yu X (2018) “Full-order terminal sliding-mode control of MIMO systems with unmatched uncertainties” Journal of franklin institute 355 (2018) 653-674.

[39] vaar S, Manimozhi M, and Enosh M (2013) “A survey of fault detection and isolation in wind turbines” 2013 international conference on power, energy and control (ICPEC).

[40] Marzat J, Piet-Lahanier H, Damongeot F, and Walter E (2011) “Model-based fault diagnosis for aerospace systems: a survey” Proc. IMechE Vol. 226 Part G: J. Aerospace Engineering.

[41] Wan l, Chen G and Sheng M (2020) “Adaptive chattering-free terminal sliding-mode control for full order nonlinear system with unknown disturbances and model uncertainties” International journals of advanced robotic systems”

[42] Boukattaya, M, Mezghani, N, and Damak, T. (2018) “Adaptive non-singular fast terminal sliding-mode control for the tracking problem of uncertain dynamical systems” ISA Trans 2018; 77(1): 1–19.

[43] Boukattaya M, Jallouli M, and Damak T (2012) “On trajectory tracking control of nonholonomic mobile manipulators with dynamic uncertainties and external torque disturbances” Robot Autonom System 2012;60(12):1640-7.

[44] Ouyang PR, Acob J, and Pano V (2014)” PD with sliding mode control for trajectory tracking of robotic system”. Robot Computer-Integrated Manufacturing 2014;30(2): 189-200.

[45] Feng Y, Yu X, and Han F (2013) “On nonsingular terminal sliding-mode control of non linear systems” Automatica 2013;49: 1715-1722.

[46] Feng Y. Yu X and Man Z (2002) “Non-singular terminal sliding mode control of rigid manipulators” Automatica 28(11) (2002) pp.2159-2167

[47] Chiu, C (2012) “Derivative and integral terminal sliding mode control for a class of MIMO nonlinear systems” Automatica, 48 (2) (2012), pp. 316-326

[48] Feng Y, Yu X, and Han F (2013) “High-order terminal sliding-mode observer for parameter estimation of a permanent-magnet synchronous motor” IEEE Transactions on Industrial Electronics, 60 (10) (2013), pp. 4272-4280.

[49] Zhao L and Jia Y. (2015) “Finite-time attitude tracking control for a rigid spdisacecraft using time-varying terminal sliding mode techniques”. Int J Control 2015; 88(6): 1150–1162.

[50] Jin M, Lee J and Ahn KK. (2015) “Continuous non-singular terminal sliding mode control of shape memory alloy actuators using time-delay estimation”. IEEE/ASME Trans Mechatron 2015; 20(2): 899–909.

[51] Yang J, Li S and Yu X. (2013) “Sliding mode control for systems with mismatched uncertainties via a disturbance observer” IEEE Trans Indus Electron 2013; 60(1): 160–169.

[52] Moreno J, Negrete D, Torres-González V and Fridman L (2016) “Adaptive continuous twisting algorithm” international journal of control, 2016 Vol. 89, no.9 1978–1806.

[53] Fengming S and Patto R (2014) “A Robust Adaptive Approach to Wind Turbine Pitch Actuator Component Fault Estimation” 2014 UKACC International Conference on Control 9th - 11th July 2014, Loughborough, U.K

[54] Li P, Hu W, and Hu R, et al. (2020) “Imbalance fault detection based on the integrated analysis strategy for variable-speed wind turbines. Int J Elect Power Energy System 2020; 116:105570.

[55] Barambones O, Cortajarena JA, and Calvo I, et al. (2019)” Variable speed wind turbine control scheme using a robust wind torque estimation”. Renew Energy 2019; 133: 354-366.

[56] Fekih A, Mobayen S, and Chen C (2021) “Adaptive Robust Fault-Tolerant Control Design for Wind Turbines Subject to Pitch Actuator Faults” Energies 2021, 14, 1791. https://doi.org/10.3390/en14061791.

[57] Gao, Z.; Liu, X. (2021) “An Overview on Fault Diagnosis, Prognosis and Resilient Control for Wind Turbine Systems. Processes 2021, 9, 300. https://doi.org/10.3390/pr9020300.

[58] Kavaz A and Barutcu B (2018) “Fault Detection of Wind Turbine Sensors Using Artificial Neural Networks” Journal of Sensors Volume 2018, Article ID 5628429, 11 pages https://doi.org/10.1155/2018/5628429

[59] Wymore M, Van Dam E, Ceylan H, and Qiao D (2015) “A survey of health monitoring systems for wind turbines,” Renewable and Sustainable Energy Reviews, vol. 52, no. 1069283, pp. 976–990, 2015.

[60] Wei X and Verhaegen M (2011) “Sensor and actuator fault diagnosis for wind turbine systems by using robust observer and filter,” Wind Energy, vol. 14, no. 4, pp. 491–516, 2011.

[61] Cho S, Gao Z and Moan T (2018) “Model-based fault detection, fault isolation and fault-tolerant control of a blade pitch system in floating wind turbines” Renewable Energy 120: 306–321.

[62] Chen W, Ding A, Haghani, Naik A, Khan Q and Yin S (2011) “Observer based FDI schemes for wind turbine benchmark” Proceedings of the 18th World Congress the International Federation of Automatic Control Milano (Italy).

[63] Freire N, Estima J and Cardoso M (2013) "Open-Circuit Fault Diagnosis in PMSG Drives for Wind Turbine Applications," in IEEE Transactions on Industrial Electronics, vol. 60, no. 9, pp. 3957-3967,

[64] You-Jin P, Shu-Kai S. Fan and Chia-Yu H (2020) “A Review on Fault Detection and Process Diagnostics in Industrial Processes” Processes 2020, 8, 1123; doi:10.3390/pr8091123 www.mdpi.com/journal/processes.

[65] Świercz M, (2015) "Signal processing methods for fault diagnostics in engineering systems," 2015 Signal Processing Symposium, 2015, pp. 1-6,

[66] Tamilselvan, P. Wang, P., Sheng, S. and Twomey, J (2013) “A two-stage diagnosis framework for wind turbine gearbox condition monitoring” Int. J. Progn. Health Manag. 2013, 4, 21–31.

[67] Meng X, Xiao L, Huang X and Chen Y (2020) “Fault-Tolerant Control of DC Microgrid Based on Non singular Terminal Sliding Mode” 2020 Chinese Automation Congress (CAC IEEE | DOI: 10.1109/CAC51589.2020.9326811

[68] Lei L, Yan-Jun L, and Shaocheng T (2019) “Neural Networks-Based Adaptive Finite-Time Fault-Tolerant Control for a Class of Strict-Feedback Switched Nonlinear Systems” IEEE TRANSACTIONS ON CYBERNETICS, VOL. 49, NO. 7,

[69] Yu X and Jiang J, (2015) “A survey of fault-tolerant controllers based on safety-related issues,” Annu. Rev. Control, vol. 39, no. 1, pp. 46–57,2015

[70] Shen Q, Jiang B, and Cocquempot V (2014) “Adaptive fuzzy observerbased active fault-tolerant dynamic surface control for a class of nonlinear systems with actuator faults,” IEEE Trans. Fuzzy Syst., vol. 22, no. 2, pp. 338–349,

[71] Deng F et al., (2017) “Energy-based sound source localization with low power consumption in wireless sensor networks,” IEEE Trans. Ind. Electron., vol. 64, no. 6, pp. 4894–4902, Jun. 2017.

[72] Phan V D, Vo C P, Dao H V and Ahn K K, (2021) "Robust Fault-Tolerant Control of an Electro-Hydraulic Actuator With a Novel Nonlinear Unknown Input Observer," in IEEE Access, vol. 9, pp. 30750-30760, 2021

[73] Abbaspour, A, Mokhtari S, Sargolzaei A, and Kang K. Y (2020) " A Survey on Active Fault-Tolerant Control Systems " Electronics 9, no. 9: 1513.

[74] Alwi, H., Edwards, C., and Tan, C.P. (2011) “Faultfeki tolerant control and fault detection and isolation” In Fault Detection and Fault-Tolerant Control Using Sliding Modes” Springer: Berlin/Heidelberg, Germany, 2011; pp. 7–27.

[75] Moor, T. (2016) “A discussion of fault-tolerant supervisory control in terms of formal languages” Annual Reviews in Control Volume 41, 2016, Pages 159-169.

[76] Fekih, A. (2015)” Fault tolerant control design for complex systems: Current advances and open research problems” In Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, pp. 1007–1012.

[77] Yu, X., and Jiang, J. (2015) “A survey of fault-tolerant controllers based on safety-related issues”. Annu. Rev. Control 2015, 39, 46–57

[78] Fekih, A. (2014) “Fault diagnosis and fault tolerant control design for aerospace systems: A bibliographical review”. In Proceedings of the 2014 American Control Conference (ACC), Portland, OR, USA, pp. 1286–1291.

[79] Abbaspour A, Sargolzaei A, Forouzannezhad P, Yen K and Sarwat A (2020) “Resilient Control Design for Load Frequency Control System Under False Data Injection Attacks," in IEEE Transactions on Industrial Electronics, vol. 67, no. 9, pp. 7951-7962, doi: 10.1109/TIE.2019.2944091.

[80] Khalid H and Peng J, (2017) "Immunity toward data-injection attacks using multi sensor track fusion-based model prediction", IEEE Trans. Smart Grid, vol. 8, no. 2, pp. 697-707.

[81] Hongbo C, and Xun W. (2010) “Fault Detection and Tolerance of Instrument Transformer Based on Substation Integrated Automation[C]”. International Conference on Electrical & Control Engineering. IEEE,2010.

[82] Corradini M, and Cristofaro A. (2018) “Non-singular terminal sliding-mode control of nonlinear planar systems with global fixed time stability guarantees”. 2018, 95:561-565.

[83] Tohidi H, Erenturk K and Shoja-Majidabad S (2017) “Passive fault tolerant control of induction motors using nonlinear block control” Control Eng. Appl. Inf., 19 (1) (2017), pp. 49-58.

[84] J. Zhu, K. Ma, A. Hajizadeh, M. Soltani and Z. Chen (2017) "Fault detection and isolation for wind turbine electric pitch system," 2017 IEEE 12th International Conference on Power Electronics and Drive Systems (PEDS), 2017, pp. 618-623.

[85] Sheibat-Othman N, Othman S, Benlahrache M, and Odgaard P (2013) “Fault detection and isolation in wind turbines using support vector machines and observers” in Proceedings of the American Control Conference.

[86] Yang W, Court R, and Jiang J (2013) “Wind turbine condition monitoring by the approach of SCADA data analysis,” Renewable Energy, vol. 53, pp. 365–376, 2013.

[87] Li M, Yu D, Chen Z, Xiahou K, Ji T and Wu Q, (2019) "A Data-Driven Residual-Based Method for Fault Diagnosis and Isolation in Wind Turbines," in IEEE Transactions on Sustainable Energy, vol. 10, no. 2, pp. 895-904, doi: 10.1109/TSTE.2018.2853990.

[88] Odgaard, P.F. and Stoustrup, J. (2009)” Unknown input observer based scheme for detecting faults in a wind turbine converter” In proceedings of the 7th IFAC symposium on fault detection, supervision and safety of technical processes, page 161–166, Barcelona, Spain, (2009).

[89] Chafouk H, and Ouyessaad H (2015) “Fault Detection and Isolation in DFIG Driven by a Wind Turbine” IFAC-Papers Online Volume 48, Issue 21, 2015, Pages 251-256.

[90] Bakri A, Boumhidi J and Boumhidi I (2019) "Support vector machine-based active fault-tolerant control for wind turbine," 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), 2019, pp. 1-5, doi: 10.1109/ICDS47004.2019.8942311

[91] Lan J, Patton R, and Zhu X, (2018) “Fault-tolerant wind turbine pitch control using adaptive sliding mode estimation”, Renewable Energy, Volume 116, Part B, pp 219-231.

[92] Edwards C, Spurgeon S, and Patton R (2000) “Sliding mode observers for fault detection and isolation” Automatica 36 (2000) 541}553

[93] Zheng X, Li He, Song R and Y. Feng (2016) "Full-order terminal sliding mode control for boost converter," 2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia), 2016, pp. 2172-2176, doi: 10.1109/IPEMC.2016.7512634.

[94] Zhou M, Feng Y and Han F (2017) "Continuous full-order terminal sliding mode control for a class of nonlinear systems" 2017 36th Chinese Control Conference (CCC), 2017, pp. 3657-3660, doi: 10.23919/ChiCC.2017.8027927.

[95] Vo A and Kang, H (2019) “Adaptive Neural Integral Full-Order Terminal Sliding Mode Control for an Uncertain Nonlinear System," in IEEE Access, vol. 7, pp. 42238-42246, 2019, doi: 10.1109/ACCESS.2019.2907565.

[96] Zhou K, Yuan C, Sun D, Jin N, and Wu X (2021) “Parameter adaptive terminal sliding mode control for Full-Bridge DC-DC converter” PLoS ONE 16(2): e0247228. https://doi.org/10.1371/journal. pone.0247228

[97] Odgaard P and Stoustrup J, (2010) "Unknown input observer based detection of sensor faults in a wind turbine," 2010 IEEE International Conference on Control Applications, 2010, pp. 310-315, doi: 10.1109/CCA.2010.5611266.

[98] Abrazeh S, Parvaresh A, Mohseni S, Zeitouni M, Gheisarnejad M, and Khooban M (2021) “Nonsingular Terminal Sliding Mode Control With Ultra-Local Model and Single Input Interval Type-2 Fuzzy Logic Control for Pitch Control of Wind Turbines” IEEE/CAA JOURNAL OF AUTOMATICA SINICA, VOL. 8, NO. 3.

[99] Blesa J, Nejjari F, Rotondo D and Puig V, (2013) "Robust fault detection and isolation of wind turbines using interval observers," 2013 Conference on Control and Fault-Tolerant Systems (SysTol), 2013, pp. 353-358.

[100] Wu Q and Saif M, (2005)"Neural adaptive observer based fault detection and identification for satellite attitude control systems," Proceedings of the 2005, American Control Conference, 2005, pp. 1054-1059 vol. 2,

[101] Tiwari P M, Chakraborty P and Soni K M, (2016) "Fault tolerant attitude control using anti-unwinding second-order sliding mode," 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), 2016, pp. 1-6,

[102] Polycarpou M M and Helmicki A J (1995) "Automated fault detection and accommodation: a learning systems approach," in IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, no. 11, pp. 1447-1458, Nov. 1995,

[103] Dávila J, Cieslak J, Henry D (2016) “Fault tolerant attitude controller for a class of additive faults via High-Order Sliding Modes” IFAC-Papers Online Volume 49, Issue 5, 2016, Pages 254-259.

[104] Schulte H, and Gauterin E (2015) “Fault-tolerant control of wind turbines with hydrostatic transmission using Takagi–Sugeno and sliding mode techniques” Annual Reviews in Control Volume 40, 2015, Pages 82-92.

[105] Wang J, Gao Y, Qiu J and Ahn C (2017) “Sliding mode control for non-linear systems by Takagi–Sugeno fuzzy model and delta operator approaches” IET control theory appl., 2017,vol. 11 Iss.8,pp.1205-1213.

[106] Xiao C, Liu Z, Zhang T, and Zhang X (2021) “Deep Learning Method for Fault Detection of Wind Turbine Converter” Appl. Sci. 2021, 11(3), 1280; https://doi.org/10.3390/app11031280

[107] Witczak M, Rotondo D, Puig V and Nejjari F (2018) “Fault estimation of wind turbines using combined adaptive and parameter estimation schemes” Int J Adapt Control Signal Process. 2018; 32:549–567.

[108] Cho S, Choi M, Gao Z, and Moan T (2021) “Fault detection and diagnosis of a blade pitch system in a floating wind turbine based on Kalman filters and artificial neural networks” Renewable Energy 169 (2021) 1-13.

[109] Xiuxing Y, Gao X, Karimi H, and Zhu X (2014) “Study on Support Vector Machine-Based Fault Detection in Tennessee Eastman Process” Hindawi Publishing Corporation Abstract and Applied Analysis Volume 2014, Article ID 836895, 8 pages http://dx.doi.org/10.1155/2014/836895

[110] Laouti N, Sheibat-Othman N, and Othman S (2011) “Support Vector Machines for Fault Detection in Wind Turbines” Proceedings of the 18th World Congress the International Federation of Automatic Control Milano (Italy).

[111] Batur C, Zhou L and Chan C (2002) "Support vector machines for fault detection," Proceedings of the 41st IEEE Conference on Decision and Control, 2002, pp. 1355-1356 vol.2, doi: 10.1109/CDC.2002.1184704.

[112] Shahid N, Aleem S A, Naqvi I and Zaffar N, (2012) "Support Vector Machine based fault detection & classification in smart grids," 2012 IEEE Globecom Workshops, 2012, pp. 1526-1531, doi: 10.1109/GLOCOMW.2012.6477812.

[113] Yusuff A, Jimoh A, and Munda L (2014) “Fault location in transmission lines based on stationary wavelet transform, determinant function feature and support vector regression” Electric Power Systems Research 110 (2014) 73–83.

[114] Lu D, Qiao W and Gong X (2017) "Current-Based Gear Fault Detection for Wind Turbine Gearboxes," in IEEE Transactions on Sustainable Energy, vol. 8, no. 4, pp. 1453-1462, Oct. 2017, doi: 10.1109/TSTE.2017.2690835.

[115] Pros and cons of hydraulic pitch systems, https://www.fsenergy.com/technology/hydraulic-vs-electric-pitch/. Fritz Schur Energy.

[116] Chen M, Shi P and Lim CC. (2016) “Adaptive neural fault tolerant control of a 3-DOF model helicopter system” IEEE T System Man Cy: S 2016; 46: 260–270.

[117] Kwan C and Lewis LF. (2000) “Robust backstepping control of nonlinear systems using neural networks," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 30, no. 6, pp. 753-766.

[118] Deng W, Yao R, Zhao H, et al. (2019)” A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm” Soft Computing 2019; 23: 2445–2462

[119] Van M, Mavrovouniotis M and Ge S, (2019) "An Adaptive Backstepping Non-singular Fast Terminal Sliding Mode Control for Robust Fault Tolerant Control of Robot Manipulators," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 7, pp. 1448-1458.

[120] Yu D, Gomm J.B, Williams D (1999) “Sensor fault diagnosis in a chemical process via RBF neural networks” Control Engineering Practice 7 (1999) 49–55

[121] Deng W, Yao R, Zhao H, Yang X, and Li G (2019) “A novel intelligent diagnosis method using optimal LS_SVM with improved PSO algorithm” soft computing 23:2445-2462. https://doi.org/10.1007/s00500-017-2940-9

[122] Omoregbee H, and Stephan P (2019) “Fault Classification of Low-Speed Bearings Based on Support Vector Machine for Regression and Genetic Algorithms Using Acoustic Emission” Journal of Vibration Engineering & Technologies (2019) 7:455–464

[123] Wang Y, (2018) "Research on the Fault Diagnosis of Mechanical Equipment Vibration System Based on Expert System," 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), 2018, pp. 636-641,

[124] Xiong J , Zhang Q , Liang Q, Zhu H, and Li H (2018) “Combining the Multi-Genetic Algorithm and Support Vector Machine for Fault Diagnosis of Bearings” Hindawi Shock and Vibration Volume 2018, Article ID 3091618, 13 pages.

[125] Yu D, Gomm J (2003) “Implementation of neural network predictive control to a multivariable chemical reactor” Control Engineering Practice 11 (2003) 1315–1323

[126] Yu D, Hamad A, Gomm J, Sangha M (2014) “Dynamic fault detection and isolation for automotive engine air path by independent neural network model” International Journal of Engine Research. 2014;15(1):87-100.

[127] Xiuxing Yin, Wencan Zhang, Zhansi Jiang, Li Pan. (2019) "Adaptive robust integral sliding mode pitch angle control of an electro-hydraulic servo pitch system for wind turbine", Mechanical Systems and Signal Processing133 (2019) 105704.

[128] Yiran Shi, Shoutao Li, Shuangxin Wang, Yujia Zhai, Yantao Tian, Ding-Li Yu. (2021) "Pitch angle control with fault diagnosis and tolerance for wind turbine generation systems", Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2021

[129] Wei X, Van Engelen T and Verhaegen M (2009) “Sensor fault detection and isolation for wind turbines based on subspace identification and Kalman filter techniques” Int. J. Adapt. Control Signal Process.2010;24:687–707.

[130] Zhang X, Zhang Q, Zhao S, Ferrari R, Polycarpou M and Parisini T (2011) “fault detection and isolation of the wind turbine benchmark: an estimation-based approach” Proceedings of the 18th World Congress. The International Federation of Automatic Control Milano (Italy).