Model-based Fault Detection and Isolation Techniques for Wind Turbines

Abdulhamed Hwas

Industrial Control Centre,
Department of Electronic and Electrical Engineering,
University of Strathclyde,
204 George Street,
Glasgow G1 1XW

Thesis submitted to the
University of Strathclyde
for the degree of
Doctor of Philosophy

August, 2013

Acknowledgments

First of all, I would like to thank my supervisor Dr. Reza Katebi, who has been an excellent mentor to me throughout the whole period of my PhD studies, and has always been available to answer my questions and give me advice. I would like also to thank Dr. Sung-ho Hur for the useful conversations on how to approach problems concerning my research.

There are not enough words to thank my parents, to whom I owe a lifetime of gratitude, for their continuous encouragement. Many thanks, too, to my wife who has constantly supported me during my studies.

Abstract

The main objective of the work of this thesis is to design model-based fault detection and isolation techniques for a large-scale wind turbine. A mathematical model of the 5MW wind turbine was developed; the model was sufficiently detailed to be used for simulation purposes. The stages of the modelling procedure were to divide the overall wind turbine system into appropriate sub-models suitable for separate modelling. Each sub-model was then presented and combined in order to obtain a completed non-linear wind turbine model. Two methods are proposed to calculate the gains of a proportional-integral (PI) pitch angle controller for the non-linear model: the first method is analytical and the second method is based on simulation. The simulation results demonstrated good performance for both proposed PI schemes. In order to design an electrical torque controller, an internal model control-based PI controller was used to find the gains of the current and of the torque controller; good static and dynamic performance were achieved.

In this thesis, a quantitative model-based method for early detection and diagnosis of wind turbine faults is proposed. The method is based on designing an observer by using a linear model of the system; the observer innovation signal is monitored to detect faults. The fault detection system was designed and optimised to be maximally sensitive to system faults and minimally sensitive to system disturbances and noise; a multi-objective optimisation method was utilised to address this dual problem. Simulation results are presented to demonstrate the performance of the proposed method. Next, a non-linear observer-based scheme for early fault detection and isolation of wind turbines was developed. The method is based on designing a non-linear observer using a non-linear model of the wind turbine. The state-dependent differential Riccati equation was used to design a non-linear observer. The comparison of system outputs with non-linear observer estimation confirmed good performance of the non-linear observer. Based on the non-linear observer, a residual generator for monitoring wind turbine model was formulated. Simulation results illustrated that the proposed method is a robust method in detecting and isolating a single fault or multi-faults in wind turbine sensors.

Table of Contents

List of Figures

List of Tables

Nomenclature

$\delta u(t)$	Actuator disturbance signal
$\Delta u(t)$	Actuator fault
$u_o(t)$	Actuator output
θ	Azimuth angle
β	Blade-pitch angle
$\delta u_c(t)$	Component disturbance signal
$\Delta u_c(t)$	Component fault
$y_0^c(t)$	Component output
ξ	Damping ratio
$W(jw)$	Dependent weighting penalties
p_i	Desired eigenvalue
$\dot{e}(t)$	Dynamic error
λ_f	Filter-tuning parameter
P_e	Generator power
θ_m	Generator rotor angle
$\xi(t)$	Input noise signal
$\overline{\sigma}\{.\}$	Maximal singular values
$\underline{\sigma}\{.\}$	Minimal singular values
ω_n	Natural frequency
$\hat{y}$	Observer output
τ_β	Pitch angle time constant
β_d	Pitch demand
ω_{m_ref}	Rated speed of the generator
ω_{wt_ref}	Rated speed of the wind turbine rotor
ε	Ratio reflecting the turbine aerodynamic efficiency
T_{e_ref}	Reference electrical torque
e_ω	Rotational speed error

ω_m	Rotational speed of the generator
ω_{wt}	Rotational speed of the wind turbine rotor
β_K	Rotor-collective blade-pitch angle at which the pitch sensitivity has doubled from its value at the rated operating point
$\frac{\partial P}{\partial \beta}$	Sensitivity of the aerodynamic power to the rotor-collective blade-pitch angle
$\Delta y(t)$	Sensor fault
$\delta y(t)$	Sensor noise signal
$\eta(t)$	Sensor noise signal
$y_o(t)$	Sensor output
σ_s	Standard deviation
ω_s	Synchronous speed
λ	Tip speed ratio
θ_{wt}	Turbine rotor angle
V_H	Wind speed on the point of blade at H
V_0	Wind speed on the point of blade at H_0
P_{wt}	Wind turbine power
$\lambda_r(j)$	Elementary tip speed ratio
σ	Leakage coefficient
A	Area swept by blades
a	Axial flow interference factor
b	Tangential flow interference factor
b_d	Damping coefficient
C_D	Drag coefficient
C_L	Lift coefficient
C_p	Power coefficient
C_s	Torsional damping
C_t	Torque coefficient
$d(t)$	Disturbance vector
dD	Elementary drag force (N)
dF	Total force acting on the blade element

dF_a	Elementary axial thrust force
dF_t	Elementary tangential force in the direction of rotation
dL	Elementary lift force (N)
$e(t)$	State error vector
f	Grid frequency
$f(t)$	Fault vector
GK	Gain-correction factor
H	Wind turbine hub height point of blade
$J_{af}(K,Q)$	Actuator performance index
$J_{cf}(K,Q)$	Component performance index
$J_d(K)$	Disturbance performance index
J_G	Generator inertia
$J_{sf}(K,Q)$	Sensor performance index
J_T	Turbine inertia
K	Observer gain matrix
K_D	Derivative gain
K_I	Integral gain
K_P	Proportional gain
K_s	Torsional stiffness
L_m	Mutual inductance
L_s	Rotor inductance
L_s	Stator inductance
N_b	Number of blades
n_g	Gearbox ratio
n_p	Pole pairs
P	Wind power
P_N	Rated turbine power
P_r	Generator rotor power
P_s	Generator stator power
Q	Residual weighting factor
R	Rotor radius
r	Weighted residual

$r_{(j)}$	Elementary distance to the hub
$R_{1,}$ R_2	Fault distribution matrices
r_a,	Actuator fault residual
r_c	Component fault residual
R_r	Rotor resistance
r_s	Sensor fault residual
R_s	Stator resistance
T_e	Electrical torque
T_{hss}	High speed shaft torque
T_i	Integral time
T_k	Threshold for the sensor number k
T_{lss}	Low-speed shaft torque
T_{mi}	Period of time to repair or replace
T_{wt}	Wind turbine torque
u	Control signal output
v	Wind speed
V_s	Stator voltage
w	Relative wind speed to the blades when the vortex motion is considered
w_0	Relative wind speed to the blades when the vortex motion is not considered
y	Output
α	Angle of attack
ρ	Air density

Abbreviations

DFIG	Doubly-Fed Induction Generator
EKF	Extended Kalman Filter
EUIO	Extended Unknown Input Observers
FDI	Fault Detection and Isolation
GA	Genetic Algorithms
HAWT	Horizontal-axis wind turbine
IMC	Internal Model Control
LHP	Left Half Plane
MOGA	Multi Objective Genetic Algorithm
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
SDC	State-Dependent Coefficients
SDDRE	State-Dependent Differential Riccati Equation
SDRE	State-Dependent Riccati Equation
UIO	Unknown Input Observer
UKF	Unscented Kalman Filter
UT	Unscented Transformation
WECS	Wind Energy Conversion System

1 Introduction

This chapter gives a brief description of the motivation and objectives of this study, research on fault monitoring and isolation for wind turbines, contributions of the thesis, organisation of thesis and publications.

1.1 Motivation

Modern wind turbines demand a high degree of automation, resulting in increasingly complex technical processes. With increased complexity, the probability of faults occurring also increases. Faults can occur in wind turbine components, sensors or actuators. Examples of common faults include: short-circuiting or overheating of electrical components, physical damage to bearings due to mechanical stresses, pipe leaks, sticking of valves and sensor drift. Faults can cause a wind turbine to operate far from its optimal operating parameters, resulting in reduced efficiency and reduced product quality; if the fault becomes sufficiently serious, complete system failure may occur resulting, in turn, in additional maintenance costs. The operation and maintenance of wind energy converters have evolved over the years, with strategies to monitor the operational condition of wind turbines being developed in response to the increasing number of wind turbines installed globally. These strategies are the breakdown, cyclic maintenance and Condition-based maintenance, each with its own advantages and disadvantages, as now described.

I – Breakdown strategy

The breakdown strategy, also known as the 'fix it when it breaks' strategy (see Figure 1.1), has been in use since before 1950 [50]. The characteristics of this strategy are that:

- components are used right up to full life time (they are repaired or replaced when they break);
- the fix time is unknown, so planning spare parts and personnel logistics is complicated; and
- the downtime (T_{down}) for offshore wind turbines is potentially long due to lack of personnel and cranes.

II- Cyclic maintenance strategy

The cyclic maintenance strategy, used since 1950, uses experience to estimate the period of time, T_{mi}, ($i=1,2...$) needed to repair or replace a new component. Figure 1.1 shows an increase in $T_{mi,}$ as a result of improved experience. The important characteristics of this method are that:

- a system is required for planning and controlling work;
- spare part and personnel logistics must be planned;
- intervals can be adjusted due to experience, for example $T_{m3}>T_{m2}> T_{m1}$, as shown in Figure 1.1;
- components are not used fully; and
- a high personnel capacity is required.

III- Condition-based maintenance strategy

Condition-based maintenance (CBM), in operation since 1970, is used for monitoring the condition of and fault presence in wind turbines.

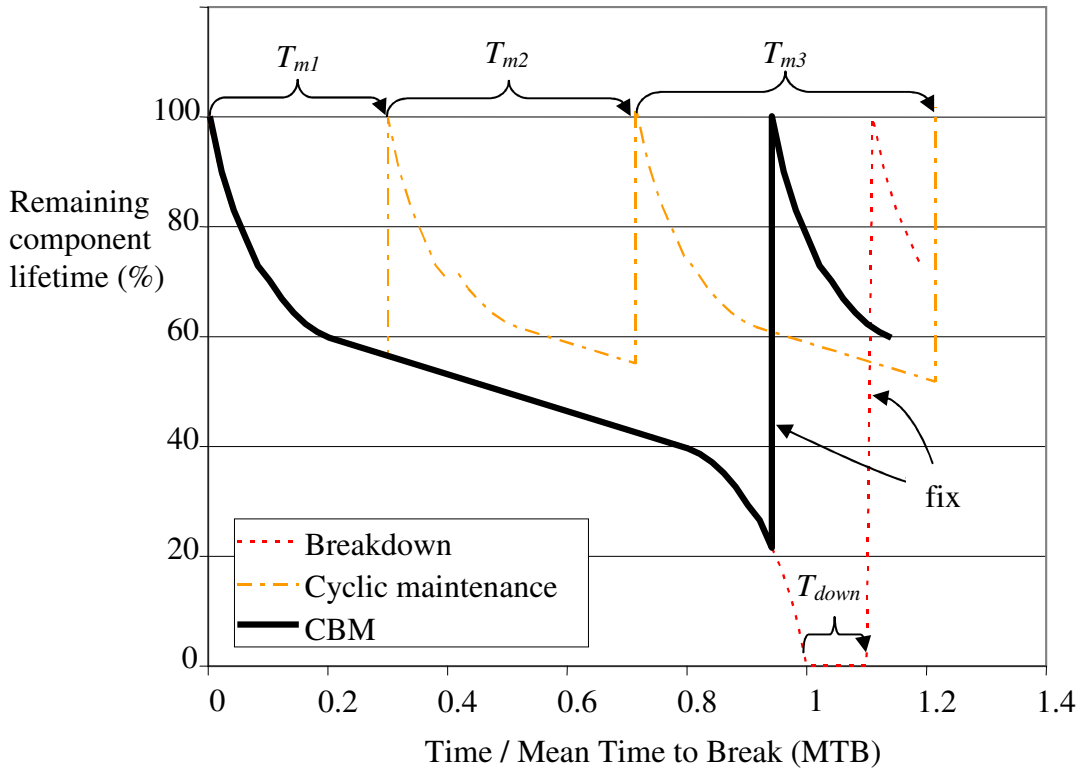


Figure 1.1: Maintenance strategies [37]

Condition monitoring achieves wind turbine reliability and maintainability. The characteristics of CBM are that:

- downtime is minimised, as shown in Figure 1.1;
- spare part and personnel logistics must be planned;
- maintenance is carried out during periods of low wind;
- a significant level of effort is dedicated to online monitoring, diagnostics etc;
- the gradient of a lifetime curve should be identified; and
- the optimal point in time for maintenance must be determined.

CBM is a preventive maintenance programme whereby a condition monitoring system (CMS), which monitors the equipment continuously, is used. The goal is to predict when a machine fault is most likely to occur. Under CBM, time intervals between maintenance operations are variable. The CBM steps are as now described.

I. Data acquisition. Online monitoring systems are employed and large volumes of data are received as measurements.
II. Data analysis and diagnosis. Improves the understanding and interpretation of fault detection.
III. Isolation. Here, a particular fault is identified and separated from other faults. For example, it may be needed to determine in which sensor, actuator or component a particular fault has occurred.

In this thesis, a robust model-based fault detection and isolation techniques to detect faults in wind turbines will be developed. They can be used to support the CMS. The task is to predict when a fault occurs, diagnose or analyse data for better understanding and interpretation of fault detection, and to isolate a particular fault from others.

1.2 **Objective of this thesis**

The main objective of this thesis is to design model-based fault detection and isolation techniques for large-scale wind turbines. Fault detection and isolation (FDI)

can be achieved by using hardware or analytical methods. Hardware-based methods have the advantages of good reliability and direct fault isolation; their disadvantages are the additional cost and space required to accommodate components and extra weight. Analytical-based methods give an indication of faults by comparing the measured and predicted outputs of the process. Analytical-based fault detection algorithms can be implemented on a computer and, hence, avoid the disadvantages related to the hardware-based fault detection methods. Analytical-based fault detection algorithms can be implemented in the same processor as the control algorithms, so that no additional hardware is required. Analytical model-based techniques use the knowledge of the monitored process and, therefore, are the most capable approaches for fault detection. These techniques include observer-based approaches, parity space approaches and parameter estimation approaches. In recent decades, observer-based methods have received considerable interest for a number of reasons: they offer rapid fault detection and require no excitation signal. Other model-based approaches, which include the parity-space approach and the parameter estimation approach, are, under certain conditions or assumptions, specific forms of the observer-based approaches [46].

In this thesis, model-based fault detection and isolation techniques for wind turbines are studied in detail. Fault monitoring for linear systems is studied extensively and is demonstrated to be extremely useful. If the process has strong non-linearities, or the operating region is too wide, the linearisation error will be too large to be handled by the linear fault detection techniques. Therefore, for large wind turbines which have non-linear dynamics, there is a necessity to study non-linear fault detection techniques.

1.3 Research on fault monitoring and isolation for wind turbines

Fault diagnosis is a key technology required to achieve higher performance and increased safety and reliability for contemporary dynamic systems. Faulty components in wind turbines can cause high losses in energy production and possible damage to the wind turbines themselves; the losses may be higher for offshore wind farms. For instance, Figure 1.2 shows the percentage breakdown of the number of

failures which occurred during the years 2000-2004 [45]. Most failures were linked to the electrical system, followed by sensors and blades/pitch components. Therefore, condition/fault monitoring should be applied to avoid abnormal event progression, to reduce productivity loss, system breakdowns and damage, all of which lead to increasing the safety and reliability of the system.

In the literature, process fault diagnoses range from analytical methods to articulate intelligence and statistical approaches. From a modelling perspective, there are methods which require accurate process models, quantitative models or qualitative models. At the same time, there are methods which do not assume any form of model information and depend only on historical process data.

Analytical fault diagnosis methods, surveyed in [19, 47, 48], can be classified into two general categories: quantitative and qualitative methods. In quantitative models, this understanding is expressed in terms of mathematical functional relationships between the inputs and outputs of a system in the form of system descriptions. For example, it may be a difference or a differential equation, a state-space model or a transfer function. In a qualitative model, the relationships are expressed in terms of qualitative functions between the different components of the system. Usually, this approach depends upon the expert's knowledge of the system in both normal and fault cases.

The type of process knowledge is the most important characteristic used in classifying fault diagnosis systems. The basic knowledge, which is needed for fault diagnosis, is the set of failures and the relationship between the observations and the failures. A diagnostic system may display them explicitly as in a lookup table or it may infer them from some source of domain knowledge. The domain knowledge may be developed from a fundamental understanding of the process using first-principles knowledge; such knowledge is known as deep, causal or model-based knowledge [51]. On the other hand, knowledge may be obtained from past experience with the process; this knowledge is known as shallow, evidential or process history-based knowledge.

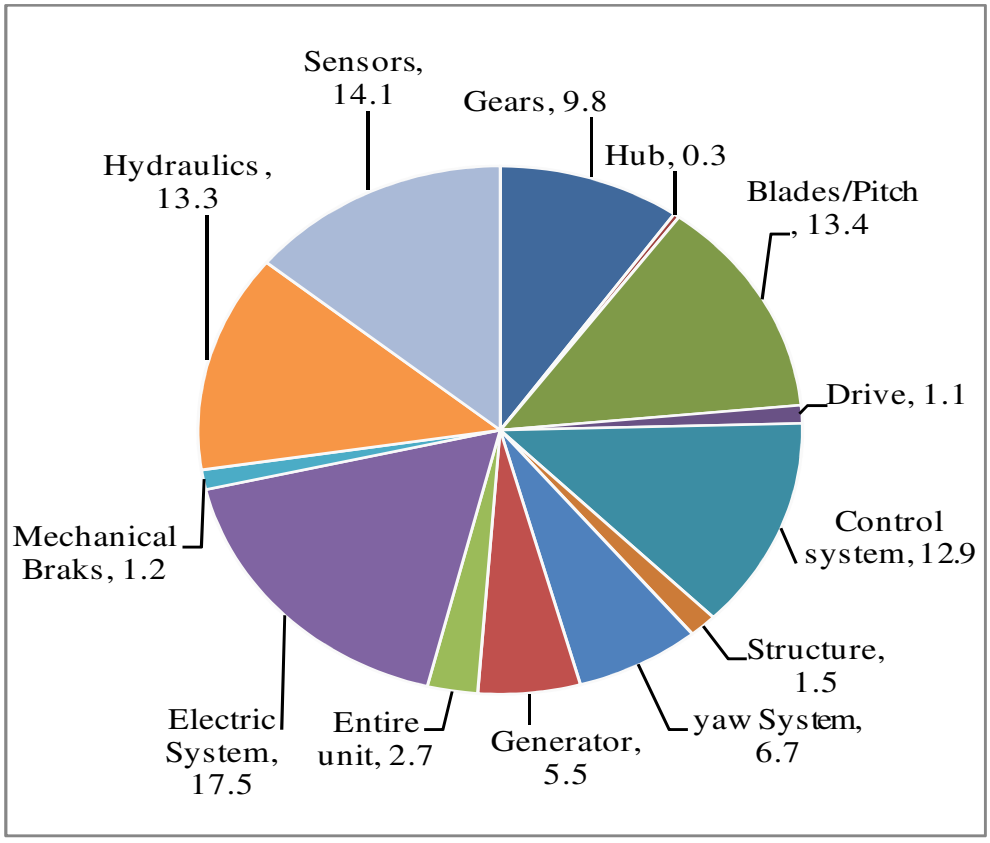


Figure 1.2: The distribution of a number of failures for Swedish wind turbines between 2000-2004 [45]

1.3.1 Basic concept of a fault

Following the SAFEPROCESS Technical Committee's suggestions, the terminologies, used in FDI, were fairly standardised. Throughout the text, a fault means an unpermitted deviation of at least one characteristic property or parameter of a system from the acceptable/usual/standard condition [52]. A particularly related term is failure which is a permanent interruption of the system's ability to perform a required function under specified operating conditions. There is a slight difference between fault and failure; failure means a component's complete breakdown, whereas fault is only a deviation from normal characteristics. As far as detection is concerned, faults and failures can be treated alike. In this thesis, the term fault is used to encompass failure.

Faults can be described as external inputs or parameter deviations which change the behaviour of the process. Like faults, disturbances and uncertainties can be modelled also as external inputs. Furthermore, disturbances and uncertainties have effects on the process, in a similar manner to the effects of faults on the process. However, compared to faults, disturbances are unavoidable and are present even during the normal operation of the process. Moreover, the controller is designed so that it can perform well in the presence of disturbances. Faults, on the other hand, are more severe changes and their effects cannot be overcome by a fixed controller and, therefore, must be detected.

The purpose of fault diagnosis is to detect faults and to determine their location and significance. The fault diagnosis procedure consists of three steps: fault detection, fault isolation and fault identification. Fault detection is the process of determining the presence of faults and the times of their occurrence. The function of fault isolation is to locate exactly the cause or the origin of the fault. Once the fault has been detected and isolated, the fault identification starts with the aim of identifying the approximate time behaviour of the fault.

1.3.2 Types of faults

A fault is defined as an unexpected deviation from the normal condition of one or more of the system's characteristic properties or parameters. Faults can be divided into three types depending on the system's components: actuators, sensors and components. Each of these faults and their effects are described briefly below.

1.3.2.1 Component faults

Component faults alter the physical parameters of the plant which, in turn, results in changes to its dynamical properties. The common reason for these faults is usually wear and tear, ageing of components etc. Some examples of component faults are leakages in oil tanks, breakages or cracks in the gearbox system and changes in friction due to lubricant deterioration etc. Component faults may result in instability of the process; therefore, it is extremely important to detect these faults.

1.3.2.2 Actuator faults

Actuators are required to transform control signals into proper actuation signals such as torques and forces in order to drive the system. The effect of a fault in an actuator may range from mere higher energy consumption to total loss of control. Examples of actuator faults include stuck-open control valves, faults in pumps, motors etc. As shown in Figure 1.3, some common actuator faults are lock-in-place, float around trim, hard-over failure and loss of effectiveness.

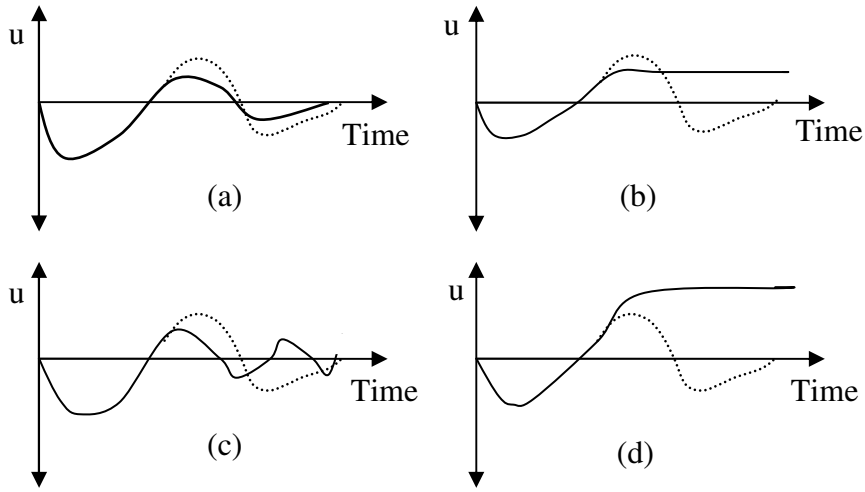


Figure 1.3: Common types of actuator faults [53]. Dotted lines show the desired value of actuator and the solid lines show actual value. (a) Loss of effectiveness, (b) lock-in-place, (c) floating around trim and (d) hard-over failure

1.3.2.3 Sensor faults

In feedback systems, the measurements obtained by sensors are used to generate the control inputs and any sensor fault can cause operating points which are far from optimal, resulting in degradation in system performance; it is therefore very important to detect these faults. As illustrated in Figure 1.4 , typical examples of sensor faults are bias, drift, loss of accuracy, sensor freezing and calibration error.

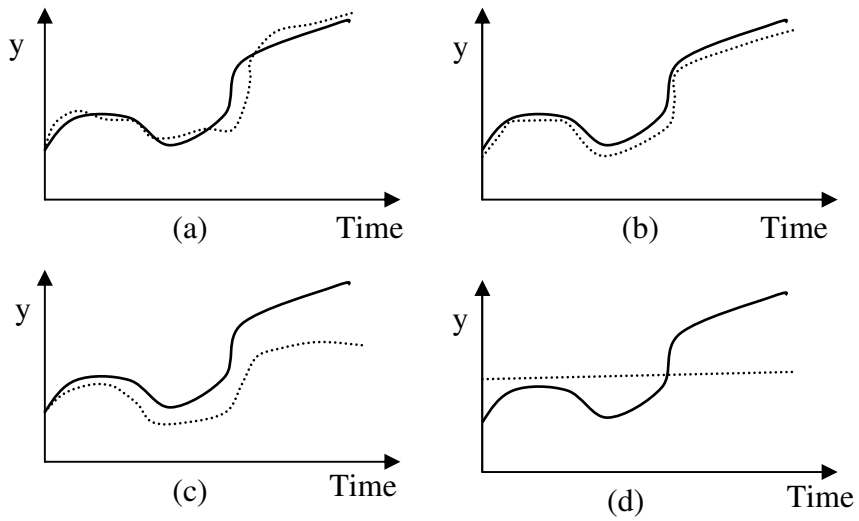


Figure 1.4 : Types of sensor faults [53]. Solid lines show the actual values whereas the dotted lines show the measured values. (a) Loss of accuracy, (b) sensor bias, (c) sensor drift and (d) frozen sensor

As shown in Figure 1.5, faults can be categorised according to whether they have developed slowly in the system (incipient faults); arisen suddenly like a step change (abrupt faults); or occurred in discrete intervals (intermittent faults). Abrupt faults have more severe effects and may result in equipment damage. However, fortunately, abrupt faults are easier to detect. Incipient faults grow slowly and result in equipment degradation. Their slowly changing behaviour makes them difficult to detect. Depending to the way in which these faults are modelled, they may be further classified into additive faults and multiplicative faults.

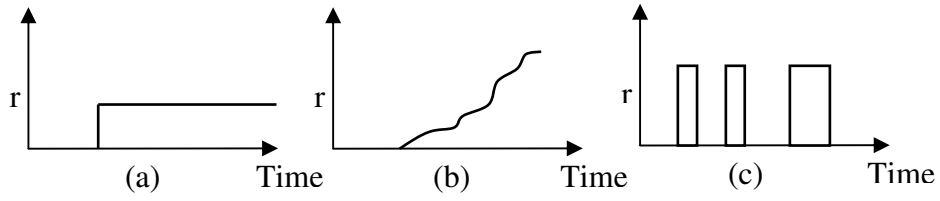


Figure 1.5: Graphical illustration of (a) abrupt fault, (b) incipient fault and (c) intermittent fault. *r* is a residual signal indicating the occurrence of the fault

1.3.3 Fault model

Faults that can be modelled as unknown changes in signals in the wind turbine system, for example changes in the control signal, sensor malfunction or slight damaged wind turbine element, are additive faults. These faults can be classified according to their source as actuator ($\Delta u(t)$), sensor ($\Delta y(t)$) and component $\Delta u_c(t)$) faults [86]. Figure 1.6 shows the effect of additive faults on the observed signals of the inputs and outputs, which are represented in equations (1.1) and (1.3). The component faults affect both the true output and the observed output, and can be written as in the equation (1.2).

$$u_o(t) = u(t) + \Delta u(t) \tag{1.1}$$

$$y_o^c(t) = u_o(t) + \Delta u_c(t) \tag{1.2}$$

$$y_o(t) = y_0^c(t) + \Delta y(t) \tag{1.3}$$

where $u_o(t)$, $y_0^c(t)$ and $y_o(t)$ are actuator, component and sensor output, respectively.

Consider additive system disturbance and sensor noise; the observed signals for the input and output can be rewritten as:

$$u_o(t) = u(t) + \Delta u(t) + \delta u(t) \tag{1.4}$$

$$y_o^c(t) = u_o(t) + \Delta u_c(t) + \delta u_c(t) \tag{1.5}$$

$$y_o(t) = y_0^c(t) + \Delta y(t) + \delta y(t) \tag{1.6}$$

where $\delta u(t)$ and $\delta u_c(t)$ are actuator and component disturbance signals, respectively and $\delta y(t)$ is sensor noise signal.

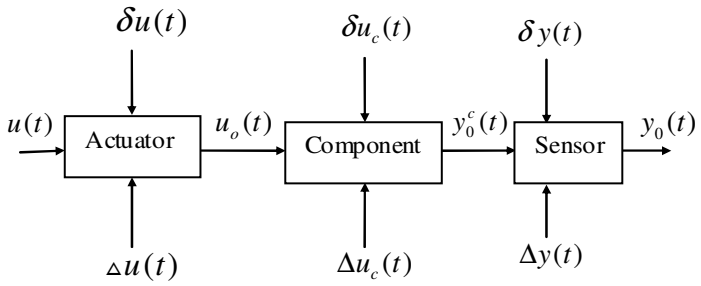


Figure 1.6: Diagram of additive faults

1.3.4 Model-based fault detection approaches

The principle of model-based fault detection schemes is to compare the behaviour of the actual process to that of the process's nominal fault-free model driven by the same input. Figure 1.7 shows the schematic diagram of a model-based fault detection scheme. It consists of two main stages: residual generation and residual evaluation. The objective of residual generation is to produce a signal, called a residual signal, by comparing the measurements with their estimates. The purpose of residual evaluation is to inspect the residual signal for the possible presence of faults.

Based on the model used for the purpose of residual generation, model-based fault detection schemes can be divided further into two categories. The model can be an analytical model, represented by a set of differential equations, or it can be a knowledge-based model represented by, for example, neural networks, petri nets, experts systems, fuzzy rules etc. Knowledge-based model approaches do not need full analytical modelling and are, therefore, more suitable in information-poor systems or in situations where the mathematical model of the process is difficult to obtain or is too complex; for example chemical processes, which are difficult to model analytically.

In analytical model-based approaches the residual signal is generated using the mathematical model of the system. The most commonly used analytical model-based approaches for residual generation are described below.

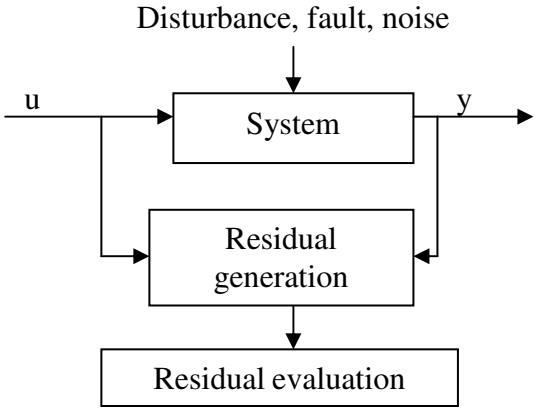


Figure 1.7: Schematic diagram of a model-based fault detection scheme

1.3.4.1 Observer-based approaches

The basic idea of the observer- or filter-based approach is to estimate the system's outputs from the measurements by using either observers in a deterministic setting [57, 58, 59, 60, 61, 62] or statistical filters (e.g. the Kalman filter) in a stochastic setting [63, 64, 65, 66]. Then, output errors, which are the difference between the system's output and the observer's output, are used as the residuals. Depending on the circumstances, linear [67] or non-linear [57, 68, 69], full or reduced-order, fixed or adaptive observers (or Kalman filters) [70, 71] may be used.

It should be noted that there is a difference between observers used for control purposes, and observers used for fault detection. The observers needed for control are state observers, i.e. they estimate states which are not measured. By contrast, output observers are the observers needed for fault detection, i.e., these observers generate estimation of the measurements.

The idea of using observers for residual generation goes back to the 1970s when Beard proposed a detection filter modified by Jones to become the so-called Beard-Jones detection filter [59]. In parallel to the Beard-Jones detection filter, the Kalman filter was used in stochastic setting. The robustness of the residual signal against unknown inputs was discussed widely in the literature and several approaches were proposed to tackle this problem. Frank et al. [72] made the first attempt to improve

the robustness of the observer-based instrument fault detection scheme. A related approach for robust residual generation is the use of the eigenstructure approach which decouples, also, the residual from unknown inputs. Compared to unknown input observers, the existing conditions for the eigenstructure assignment approach are more relaxed. In this approach, instead of decoupling state estimations from unknown inputs, the residual signal is made independent of unknown inputs. For example, the effective wind speed on the blades of a wind turbine is an unknown input.

1.3.4.2 Parity space approaches

Parity equations are rearranged and, usually, are transformed variants of the system's input-output or state-space models [73]. The basic idea is to check the parity of the system models with sensor output measurements and known process inputs. The purpose of this approach is to rearrange the model structure in order to obtain the best fault isolation. Willsky [64] introduced dynamic parity relations. Redundancy provides freedom in the design of residual generating equations so that further fault isolation can be achieved. Fault isolation requires the ability to generate residual vectors which, for different faults, are orthogonal to each other. Gertler et al. [74, 134] suggested a so-called orthogonal parity equation approach in designing structured residual sets. The design of directional residual vectors, using parity relations, is not straightforward. The systematic approaches of designing parity equations with directional properties are presented in [75, 134]. Chow and Willsky [76] proposed a procedure to generate parity equations from the state-space representation of a dynamic system.

1.3.4.3 Parameter estimation-based approaches

The parameter estimation approach for fault detection was proposed first in [77, 78] and is based on the assumption that the faults are reflected in the physical parameters of the system. With this assumption, the parameters of the system are estimated online repeatedly; any discrepancy between the estimated and the actual parameters gives an indication of the faults. An advantage of the parameter estimation approach

is that, with only one input and one output signal, several parameters can be estimated, giving a detailed picture on internal process quantities [79]. Another advantage of this method is that it yields the size of the deviations, which is important for fault analysis [55]. Although it can also detect sensor and actuator faults, a parameter estimation-based approach is useful for component fault detection. A disadvantage is that an excitation is needed in order to estimate the parameters; this may result in problems if the process is operating at stationary points [55]. There are several parameter estimation techniques including methods of least squares, recursive least squares, extended least squares, etc. Parameter estimation techniques have been applied, also, to fault detection in non-linear systems. A study of parameter estimation-based fault detection in non-linear systems can be found in [79] and an application to a non-linear satellite model in [78].

1.3.4.4 Residual evaluation methods

Residual evaluation is the second step in a model-based fault detection scheme. In this step, the residual signal is manipulated to indicate the occurrence of the fault. In ideal situations, when there are no disturbances or their effect on the residual signal is eliminated completely, there are no modelling uncertainties and the observer's initial conditions are the same as those of the process. In a fault-free case, the residual signal is zero; any deviation of the residual from zero indicates the presence of faults. However, these ideal situations are never attained and there are always modelling errors; the observer's initial conditions may be different from those of the process. Even in the absence of faults, this causes the residual signal to deviate from zero. Even in the presence of disturbances and uncertainties, the purpose of residual evaluation is to decide about the occurrence of faults.

Residual evaluation consists of three stages: residual processing, threshold selection and decision making. Two strategies have been used for residual processing. For deterministic systems, a norm-based residual processing strategy is preferred and, for stochastic systems, statistical methods are adopted. Other commonly used evaluation functions for deterministic systems include absolute value, peak value, average

value, moving average etc. In stochastic settings, the frequently used evaluation functions are mean, variance, likelihood ratio, generalised likelihood ratio etc. Further details about stochastic evaluation functions can be found in [81].

The selection of threshold is the second stage in residual evaluation. If the selected threshold is too low, it results in false alarms, i.e. some disturbances will cause the residual to cross the threshold and result in an alarm. If the selected threshold is too high, small faults are undetected. A detailed study of different threshold selection methods and their computation details for linear systems can be found in [81]. Although the residual generation for non-linear systems has been studied extensively, little attention has been given to threshold computations for non-linear systems [82, 83, 84]. Usually, in deterministic settings, the selected threshold is slightly higher than the premium value of the evaluated residual signal in a fault-free case.

Decision logic is the third stage in residual evaluation. The simplest decision logic is to compare the evaluated residual signal with the threshold. If the evaluated residual exceeds the threshold then the fault alarm is released. There are, also, some approaches which use fuzzy logic or neural networks for residual evaluation [55].

1.3.4.5 Selection of fault detection method

Selection of a suitable fault detection scheme is difficult and the decision upon which the fault detection method should be used depends on several factors (see Figure 1.8) including the availability of the mathematical model, information about the process, type of disturbances and uncertainties, non-linearities, the presence of closed or open loop etc. For example, in electrical and mechanical systems it is relatively easy to obtain a mathematical model. Consequently, quantitative model-based approaches are preferred and, usually, these are faster and their online implementation is easier making them more suitable for processes with fast dynamics. By contrast, chemical and industrial processes are difficult to model and complex, even when a mathematical model can be obtained; in such cases qualitative model-based approaches should be applied.

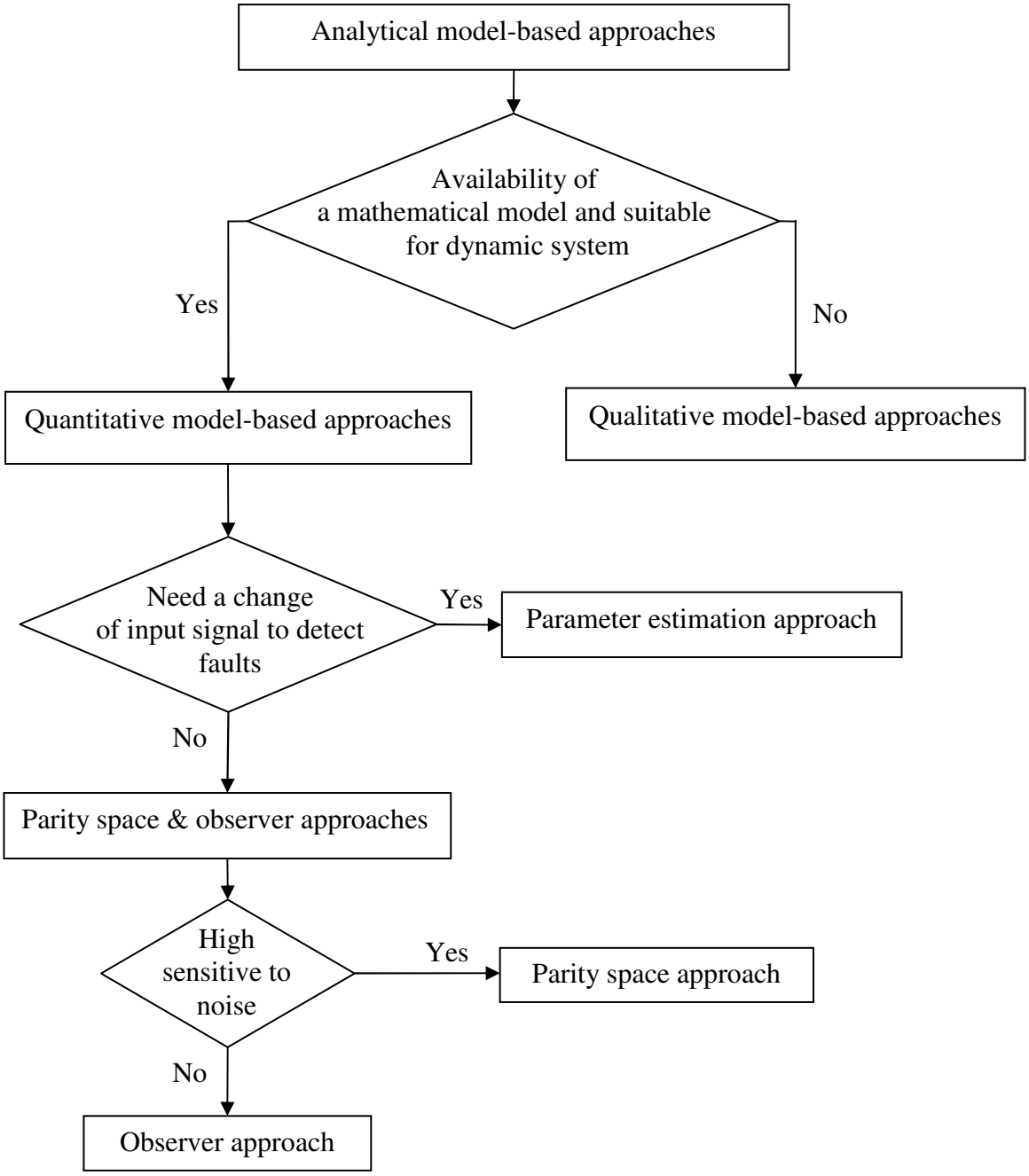


Figure 1.8: Flowchart describing the selection of a fault detection method for a wind turbine

Although the parameter estimation-based approach can also detect sensor and actuator faults, it is useful in detecting component faults. Neither the parity space

approach nor the observer-based approach need a change of input signal to detect faults, whereas a parameter estimation approach requires input excitation. Compared to observer-based approaches and parameter estimation approaches, parity space approaches are more sensitive to measurement noise [79].

Several researchers have observed a degree of correspondence between observer-based and parity relation approaches when the observer has been designed as a dead-beat. A full derivation of the equivalence of this structure can be found in [46]. The inconvenience of using dead-beat observers in the presence of noise is well known. However, for instance, a Kalman observer can be used to minimise the effect of noise in the estimation error. This leads to thinking that the resulting residual generated by an observer scheme can be superior to that obtained under certain conditions with parity relations.

1.4 Contributions of the thesis

The contributions of the thesis are now listed.

a) A model of the 5MW wind turbine is developed. The stages of the modelling procedure are to divide the overall wind turbine model into appropriate sub-models suitable for separate modelling. These sub-models are rotor torque, drive-train, doubly-fed induction generator and fixed-point wind speed model and controller. The sub-models are then combined to obtain a completed non-linear wind turbine model.

b) Two PI-based control design schemes are proposed for the control of blade-pitch angle in variable-speed wind turbines. The first method is analytical and the second method is based on simulation to calculate PI gains.

c) A model-based linear fault detection and isolation scheme for wind turbines is developed. The method is based on designing an observer using a model of the wind turbine. The fault detection system is designed and optimised to be

maximally sensitive to system faults and minimally sensitive to system disturbances and noise; a multi-objective optimisation method is employed to address this dual problem.

d) A novel non-linear observer-based fault detection and isolation method is developed. The non-linear observer is designed using the state-dependent differential Riccati equation rather than algebraic Riccati equations, which can require an overly restrictive requirement on system observability and controllability. The fault detection system is designed and optimised to be maximally sensitive to system faults and minimally sensitive to system disturbances and noise. The residual generator based on the non-linear observer is employed to develop a fault monitoring system. Simulation results demonstrate that this method is robust for the detection and isolation of a single fault or multi-faults in wind turbine sensors.

1.5 Organisation of the thesis

The organisation of the thesis is now described.

Here, in Chapter One, the motivation, objective, outline and contributions of the thesis are presented.

Chapter Two sets up a mathematical model of the large-scale wind turbine and fixed-point wind speed model. These models are sufficiently detailed to be used as a simulation model. The simulation is implemented in three steps for assessing the performance of these models. Firstly, the response of the wind turbine rotor's open loop is tested. After that, the performance of the wind turbine system is examined in cases of constant wind speed and non-stationary wind speed.

Chapter Three presents a PI controller design for controlling electrical torque and pitch angle. For a non-linear model of a 5MW wind turbine two methods are proposed to calculate the gains of a PI pitch angle controller; the first method is analytical and the second is based on simulation. In order to design an electrical

torque controller, an internal model control (IMC)-based PI is used to find the gains of the current and torque controller to achieve good static and dynamic performance.

Chapter Four describes the basic concepts of a fault and the types of faults, and presents the most common analytical model-based approaches for fault detection. Linear observer-based fault detection and isolation methods for a wind turbine are proposed. Firstly, an observer-based residual generator is designed using a set of structured residual generators. Secondly, a robust observer-based sensor fault detection and isolation scheme is developed to accomplish fault detection and isolation. Here, each sensor residual is separated from the output of the residual and the dimension of the residual is modified using a residual weighting factor.

This scheme is systematic and easy to design and implement. Simulation results demonstrate that it is suitable for detection and isolation of faults in sensors and that it is simple to handle multiple faults.

Chapter Five presents a survey of methodologies in the theory of a non-linear observer-based fault diagnosis. Those methods which this author considers suitable for application in monitoring wind turbines and which might gain some relevance for future research and practical applications are given a focus. A novel non-linear observer-based fault detection and isolation method for monitoring wind turbines is presented. Different additive sensor faults are used to test the proposed method. These faults are pitch angle, the difference between turbine rotor angle and generator rotor angle, wind turbine rotor speed, generator rotor speed, electric torque and wind turbine torque. A dynamic threshold is designed for each sensor to identify the fault based on residual curves, with consideration given to the accuracy of each sensor; in the case of no sensor fault, unity (1) is returned, and in the case of a sensor fault, zero (0) is returned. Simulation results demonstrate that it is a robust method for the detection and isolation of wind turbine faults.

Finally, Chapter Six presents the conclusions of the work of this thesis and makes recommendations for future work concerning non-linear observer-based fault diagnosis methods and condition monitoring of wind turbines.

1.6 Publications

1) Hwas, A. and Katebi, R., "*Wind Turbine Fault Monitoring"*, Tutorial Training workshop improved control of wind farms", Advanced control for offshore wind-farms to reduce failures & maintenance costs, 25th-26th May 2011, Glasgow.
2) Hwas, A. and Katebi, R. "*Wind turbine control using PI pitch angle controller"*, In IFAC Conference on Advances in PID Control PID'12, 28-30 March 2012, Italy.
3) Hwas, A., Katebi, R., "*Model-based fault detection and isolation for wind turbine"*, UKACC International Conference on Control (CONTROL). IEEE Conference Proceedings, pp. 876-881, 3-5 Sept. 2012, Cardiff.
4) Hwas, A., Katebi, R., "*Nonlinear observer-based fault detection and isolation for wind turbines" (*" (Submitted to the International Renewable Energy Congress (IREC), on March 25-27, 2014, Tunisia).
5) Hwas, A., Katebi, R., "*Linear and Nonlinear Observer-based Fault Detection and Isolation Methods for Wind Turbine"* (Journal paper, submitted to Wind Energy, October 2013).

2 Wind Energy Conversion System Models

The purpose of this chapter is to set up a mathematical model of large-size wind turbines; the model will be sufficiently detailed to be used as a simulation model. The stages of the modelling procedure are to divide the overall wind turbine model into appropriate sub-models suitable for being modelled separately. These sub-models are: rotor torque model, drive-train model, doubly fed induction generator model, grid-side converter model and fixed-point wind speed model and controller. Each wind turbine sub-model is presented and combined to obtain a complete non-linear model of a wind turbine.

2.1 Wind turbine overview

Wind turbines are categorised into two types depending on the axis of the turbine. The most common type, the horizontal axis wind turbine (HAWT), has a turbine that rotates around a horizontal axis. Less common is the vertical-axis wind turbine, where the rotor rotates around a vertical axis.

HAWT's are usually three-bladed (cyclic loads on the turbine shaft for three-bladed rotors are much smaller than those produced by two-bladed rotors [9]), with the main rotor shaft and electrical generator at the top of the tower. The rotor of a HAWT must be pointed into the wind. Small HAWT's are pointed by a tail and large HAWT's use a wind sensor coupled with a servo motor. Most wind turbines have a gearbox, which turns the slow rotation of the blades into a faster rotation that is more suitable to drive an electrical generator. Since turbulence is produced behind the turbine tower, it is usual for the turbine to be pointed upwind of the tower and thus, avoid the turbulent area. The terms 'downwind rotor' and 'upwind rotor' denote the location of the rotor with respect to the tower and wind direction (see Figure 2.1). Downwind turbines do exist, despite the problem of turbulence, since they do not require an additional mechanism to keep them facing into the direction of the wind. Furthermore, in cases of high wind speed, the blades of downward turbines can be allowed to bend, thus reducing their sweep area [3].

Modern HAWT's are large and, rather than being fixed-speed, they are variable-speed and pitch-controlled, they have a drive-train with or without gearboxes and they are highly controllable. The variable-speed operation of HAWT's allows the rotational speed of the turbine to be adapted continuously in such a manner that the turbine operates constantly at its optimal level of aerodynamic efficiency. While fixed-speed wind turbines are designed to achieve maximum aerodynamic efficiency at a single wind speed, variable-speed wind turbines achieve maximum aerodynamic efficiency over a wide range of wind speeds. Furthermore, variable speed operation allows the use of advanced control methods to increase power capture [1, 34].

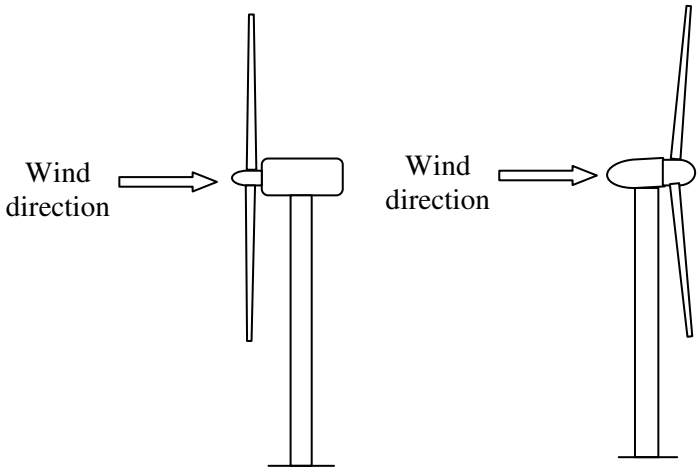


Figure 2.1: Schematic view of upwind (left) and downwind (right) of HAWTs

2.2 Modern horizontal-axis wind turbines: Main components

The main components of a modern HAWT are illustrated in Figure 2.2 and described below.

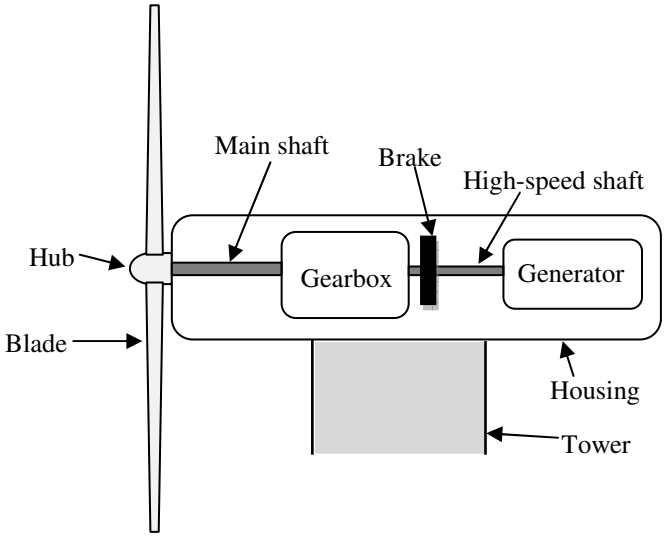


Figure 2.2: Main components of a HAWT

2.2.1 Rotor

The rotor consists of three blades that are fastened to the central hub (see Figure 2.2). It has a mechanism to adjust the blade-pitch angle (β), which is the angle between the blade chord line and the plane of the rotation. This pitch change mechanism, which controls the rotational speed of the rotor, consists of high quality mechanical devices with actuators and computerised controls.

2.2.2 Drive-train

The HAWT drive-train consists of a main shaft (known as a low-speed haft or turbine shaft), a gearbox, a high-speed shaft, a generator, a rotor break and auxiliary equipment for control, lubrication and cooling systems. Some HAWTs have a direct drive from the turbine to the generator with no gearbox, turbine shaft or drive shaft.

The low-speed shaft with its bearings, lubrication and couplings, it may include rotor control and safety devices such as sensors and a rotor brake, as well as rotary hydraulic couplings, slip rings for power and data transfer, and attached electrical and mechanical equipment with their necessary cables and piping.

Rotor torque oscillations can be attenuated before they reach the gearbox by applying the correct amount of torsion in the turbine shaft. However, the turbine shaft also needs to be strong under deformation in order to support the rotor weight. To meet these requirements, the turbine shaft may be composed of two concentric shafts

connected at the hub end and separated at the gearbox end; the outer shaft is a stiff primary structure supporting the rotor and the flexible inner shaft, the so-called quill shaft, transmits only torque to the gearbox [35].

The HAWT gearbox has a step-up ratio equal to the generator shaft speed divided by the turbine shaft speed; this ratio varies from 1 to 100 in a large scale HAWT. The step-up ratio is determined by generator design, rotor diameter and blade tip speed.

A high-speed shaft is a conventional machine element with bolted flanges on both ends. The rotor brake disk may be mounted on the drive shaft rather than on the turbine shaft for multiplication of the braking power to the square of the step-up ratio. Small-scale wind turbines may drive variable speed alternators and direct current (DC) generators, whereas medium- and large-scale HAWTs use alternating current (AC) generators.

2.2.3 Tower

The tower of the wind turbine carries the nacelle (housing) and the rotor. Most large wind turbines have tubular steel towers, which are manufactured in sections with flanges at either end and bolted together on-site. The towers are conical, with their diameter increasing towards the base in order to both increase the tower strength and minimise material use.

2.3 Wind turbine models

Wind turbine modelling is the focus of contemporary research by many academic institutions and companies. Although there are many publications in the literature, clarifications, extensions and adaptations are still needed for wind turbine models.

Published simulation and analysis information on modelling wind turbine presents various modelling approaches. Often, electrical engineers attempt to simplify the aerodynamic, mechanical parts and stress generator descriptions. In contrast, mechanical engineers disregard generator performance details. Therefore, some published models lack a detailed description of their parameters.

This section presents aerodynamic modelling of wind turbines. The aim is to present wind turbine model structures that allow the prediction of the output of a wind turbine during continuous operation. Here, a doubly-fed induction generator model for a variable-speed turbine is employed.

2.3.1 Theoretical wind turbine model

A model for the entire wind energy conversion system (WECS) can be structured as several interconnected subsystem models, as shown in Figure 2.3. The model encompasses the rotor, drive-train and the generator. The rotor subsystem describes the transformation of the three-dimensional wind speed field into aerodynamic torque that originates from the rotational movement. The drive-train transfers the aerodynamic torque on the blades to the generator shaft. The electrical subsystem describes the conversion of mechanical power at the generator shaft into electricity. Finally, the actuator subsystem models the behaviour of the pitch servo.

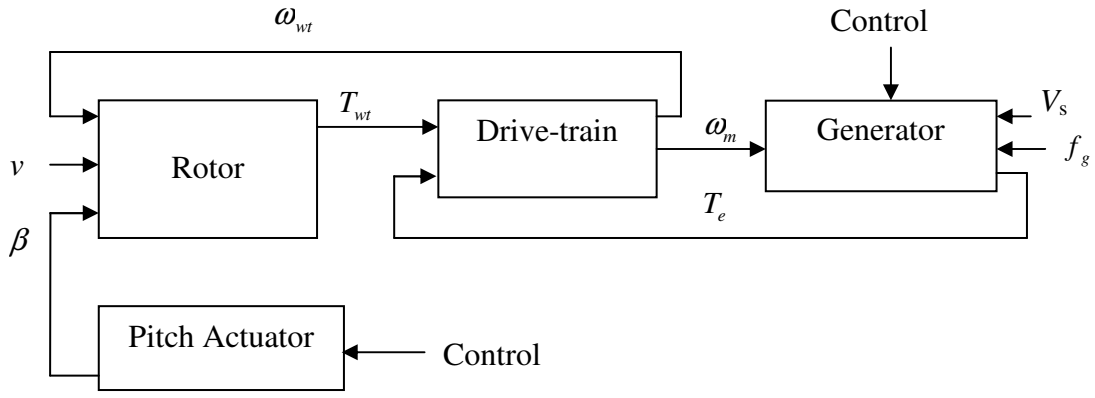


Figure 2.3: Block diagram of a variable-speed variable-pitch WECS [4]

2.3.1.1 Variable pitch blades

As shown in equation (2.1), the primary aim of modelling is to determine the wind torque generated by the turbine rotor in the form:

$$T_{wt} = f(\beta, v, \omega_{wt}) \tag{2.1}$$

where:

v : wind speed (m/s)

ω_{wt} : rotational speed of the rotor (rad/s)

β : blade-pitch angle (degrees)

Based on blade element theory, the blade can be divided into a number of cross-sectional elements along its length. A blade element *j* is obtained by sectioning the blade with two parallel planes located at distances *r* and (*r* + *dr*) from the hub and perpendicular to the blade. Figure 2.4 shows the blade element profile and the undertaken aerodynamic load, where:

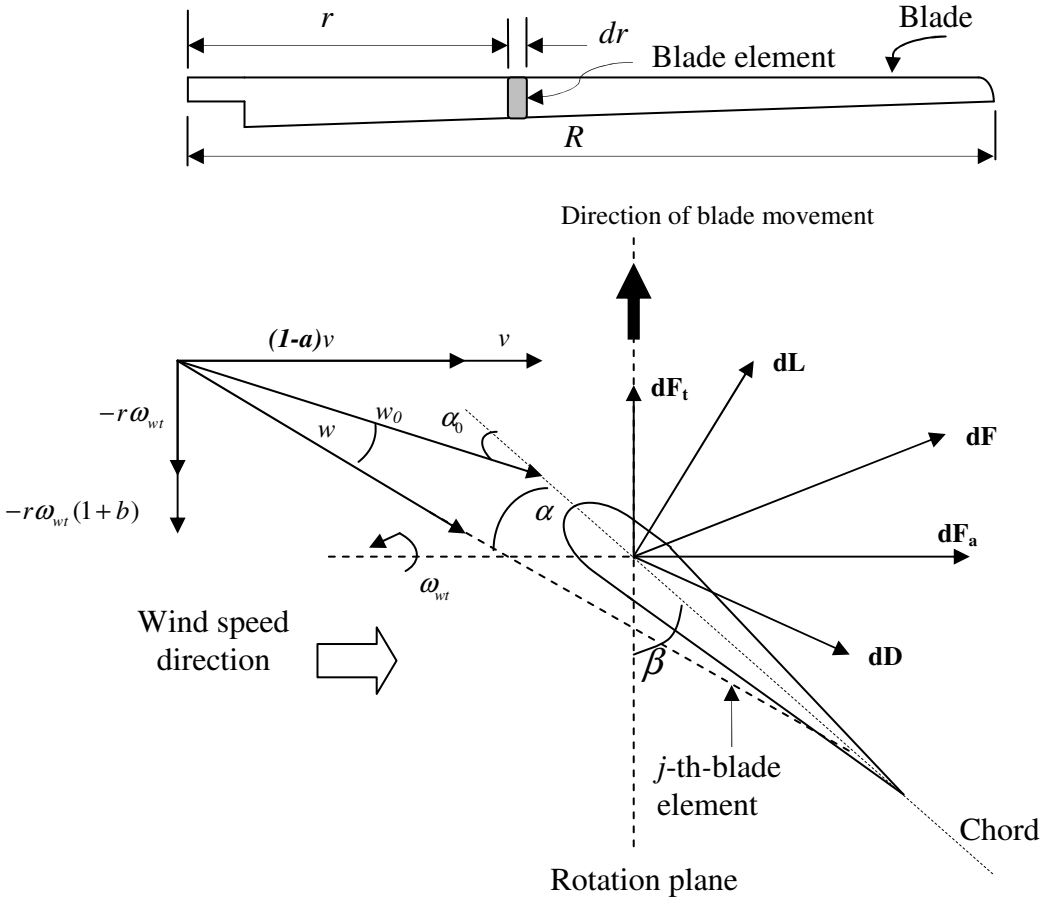


Figure 2.4: Aerodynamic loads along the blade profile

w : relative wind speed to the blades when the vortex motion is considered (m/s)

w_0 : relative wind speed to the blades when the vortex motion is not considered (m/s)

a : axial flow interference factor

b : tangential flow interference factor

dF : total force acting on the blade element (N)

dD : elementary drag force (N)

dL : elementary lift force (N)

dF_t : elementary tangential force in the direction of rotation (N)

dF_a : elementary axial thrust force (N)

The main modelling assumptions of the blade elementary theory are that [9]:

- interactions between adjacent elements of the same blade are disregarded;
- the radial component of the speed is disregarded;
- the aerodynamic coefficients are a function of the angle of attack and the blade profile; and
- infinite number of blade sections is assumed.

In turbine motion, the basic concept is that the wind flow has a relative movement to the rotating blade, defined by the relative wind speed to the blades (*w*). This variable has angle of attack (α) with the blade element reference chord. This is an important variable in determining the aerodynamic behaviour of the turbine. It is possible for the angle of attack to be changed by wind speed, rotational speed and pitch angle.

The elementary axial and tangential interference factors are computed based on the tip speed ratio $\lambda_r(j) = \omega_{wt} r(j) / v$ and on the Lagrange coefficient $K_{Lc} = 0.5\pi\rho R^2$ (is considered invariant) to find the approximate values for *a(j)* and *b(j)* [10]:

$$a(j) = \frac{K_{Lc}}{(1-K_{Lc})^2} \cdot \frac{\lambda_r^2(j)}{1+\lambda_r^2(j)/(1-K_{Lc})^2} \tag{2.2}$$

$$b(j) = \frac{K_{Lc}}{(1-K_{Lc})} \cdot \frac{1}{1+\lambda_r^2(j)/(1-K_{Lc})^2} \tag{2.3}$$

From Figure 2.4, tan ($\alpha(j)+\beta(j)$) is given by:

$$\tan(\alpha(j)+\beta(j)) = \frac{(1-a(j))v^2}{r^2(j)\omega_{wt}^2(1+b(j))} \tag{2.4}$$

Thus, the angle of attack of the *j*-th element is given by:

$$\alpha(j) = \tan^{-1}\left(\frac{1-a(j)}{\lambda_r^2(j)(1+b(j))}\right) - \beta(j) \tag{2.5}$$

where $\lambda(j)$ is the elementary tip speed ratio $\lambda(j) = r(j)\omega_{wt} / v$.

The wind speed relative to the blade is deduced from the speed diagram (Figure 2.4):

$$w(j) = v\sqrt{(1-a(j))^2 + \lambda_r^2(j)(1+b(j))^2} \tag{2.6}$$

The elementary lift force in the direction of rotation is computed as:

$$dF_t(j) = 0.5\rho\, c(j)w^2(j)C_L(\alpha)\sin(\beta(j)+\alpha(j))[1-\varepsilon\, c\tan(\beta(j)+\alpha(j))].dr \tag{2.7}$$

where c is the chord, $C_L(\alpha)$ is the lift coefficient, $C_D(\alpha)$ is the drag coefficient and $\varepsilon(\alpha)$ is a ratio reflecting the turbine aerodynamic efficiency. As shown in equation (2.8) [11].

$$\varepsilon(\alpha) = \frac{C_L(\alpha)}{C_D(\alpha)} \tag{2.8}$$

Each blade element develops an elementary torque $dT(j)$ given by:

$$dT(j) = r(j).dF_t(j) \tag{2.9}$$

where *r(j)* is the elementary distance to the hub *r(j)* and $dF_t(j)$ is the elementary tangential force in the direction of rotation.

By integrating equation (2.9) along the blade length and by using equation (2.7), the total torque is given by:

$$dT(j) = \int r(j).dF_t(j) \tag{2.10}$$

2.3.1.2 Fixed-pitch blades

The torque of the fixed-pitch wind turbine depends on the rotational speed of the low-speed shaft and the wind speed. It is possible to develop more complicated aerodynamic models which consider the rotational sampling filter, spatial filtering or induction lag, leading to a more complex expression of the developed torque [3, 12, 13]. The rotational sampling is due to the rotating motion of the blades inside the turbulence field of the fixed-point wind speed [5]. The spatial filter averages the variations in wind speed across the area swept by the rotor. The induction lag describes an aerodynamic effect demonstrated experimentally when the wind speed or the pitch angle changes. In addition, power coefficient characteristics take into account the effects arising from the Reynolds number and air density changes. For low power wind turbines these effects, together with the structural dynamics, can be disregarded and simplified models can be used. In the case of a fixed-pitch wind turbine, the torque (T_{wt}) is given by:

$$T_{wt} = f(v, \omega_{wt}) \tag{2.11}$$

The aerodynamic performance of a wind turbine is usually characterised by the variation of the non-dimensional C_P *vs.* λ curve and the power extracted by a wind turbine is expressed as [9]:

$$P_{wt} = 0.5\rho\pi R^2 v^3 C_p(\lambda) \tag{2.12}$$

The equation of the wind turbine torque can be written as:

$$T_{wt} = \frac{P_{wt}}{\omega_{wt}} = 0.5\ \pi\rho v^2 R^3 C_t(\lambda) \tag{2.13}$$

where the torque coefficient is given by:

$$C_t = \frac{C_P}{\lambda} \tag{2.14}$$

Figure 2.5 shows a block diagram for the wind torque, which illustrates a simplified description of the wind speed fluctuations due to wind turbine interaction [12]. The fixed-point wind speed model is described and discussed in section 2.5. The target of the spatial filter is to perform the variations of average wind speed across the area swept by the rotor. The transfer function of the spatial filter is represented in equation (2.15) [12, 14]:

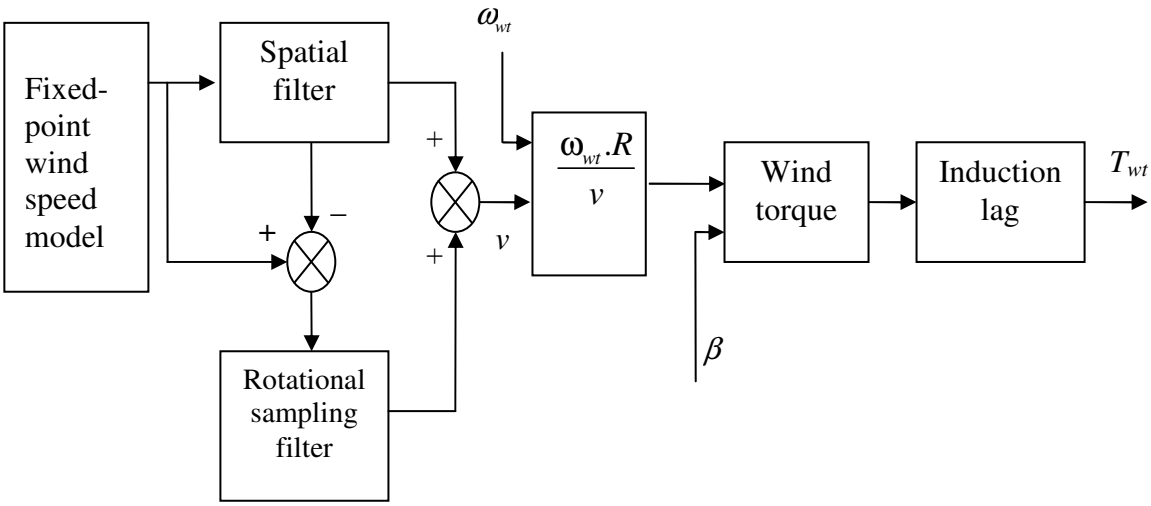


Figure 2.5: Block diagram of wind torque

$$H_{sf}(s) = \frac{\sqrt{2} + b_{sf}\ s}{(\sqrt{2} + b_{sf}\sqrt{a_{sf}}\ \ s)(1 + \frac{b_{sf}}{\sqrt{a_{sf}}}s)} \tag{2.15}$$

where:

a_{sf} : empirical factor = 0.55

b_{sf} : parameter describing the relation between wind speed evolutions in different points across the rotor, which can be expressed as:

$$b_{sf} = \gamma_{sf}(\frac{R}{v_s}) \tag{2.16}$$

with $\gamma_{sf} = 1.3$

R : wind turbine radius

v_s : average wind speed at the hub

According to Wilkie et al. (1990) and Rodriguez-Amenedo et al. (1998) [12, 14], the transfer function of the rotational sampling filter is erpresented by equation (2.17):

$$H_{rs}(s) = \frac{(s + N_b\omega_{wt} + \varepsilon)(s + N_b\omega_{wt} - \varepsilon)}{(s + \sigma_s)^2 + (N_b\omega_{wt})^2} \tag{2.17}$$

where parameters ε and σ_s determine the magnitude of the power density concentration at N_b and ω_{wt}. N_b is the number of blades [12, 14].

In addition, Figure 2.5 depicts the subsystem of the induction lag, which is described as a transient aerodynamic effect proven experimentally when the wind speed or the pitch angle changes suddenly. This effect is modelled as a lead-lag compensation and is presented in equation (2.18) [14]:

$$H(s) = \frac{c\,s + 1}{d\,s + 1} \qquad c > d \tag{2.18}$$

where the time constants *c=11.25s* and *d=7.5s.*

2.4 Model of a 5MW wind turbine

In the development of wind turbine technology several types of wind turbines such as fixed-speed, variable-speed wind turbines and wind turbines with doubly-fed induction generator have been constructed and tested. Recently, wind turbines with a doubly-fed induction generator (DFIG) have become more acceptable due to their characteristics of high energy transfer efficiency, low investment and flexible control. Therefore, in the following sections, mathematical models of wind turbine with DFIG were studied and implemented, as shown in Figure 2.6.

This section is concerned with the modelling of a 5MW wind turbine in two modes for both under-rated and over-rated wind speed. These modes are variable-speed fixed-pitch and fixed-speed variable-pitch (see nominal physical parameters of the 5MW wind turbine in Appendix A [16]).

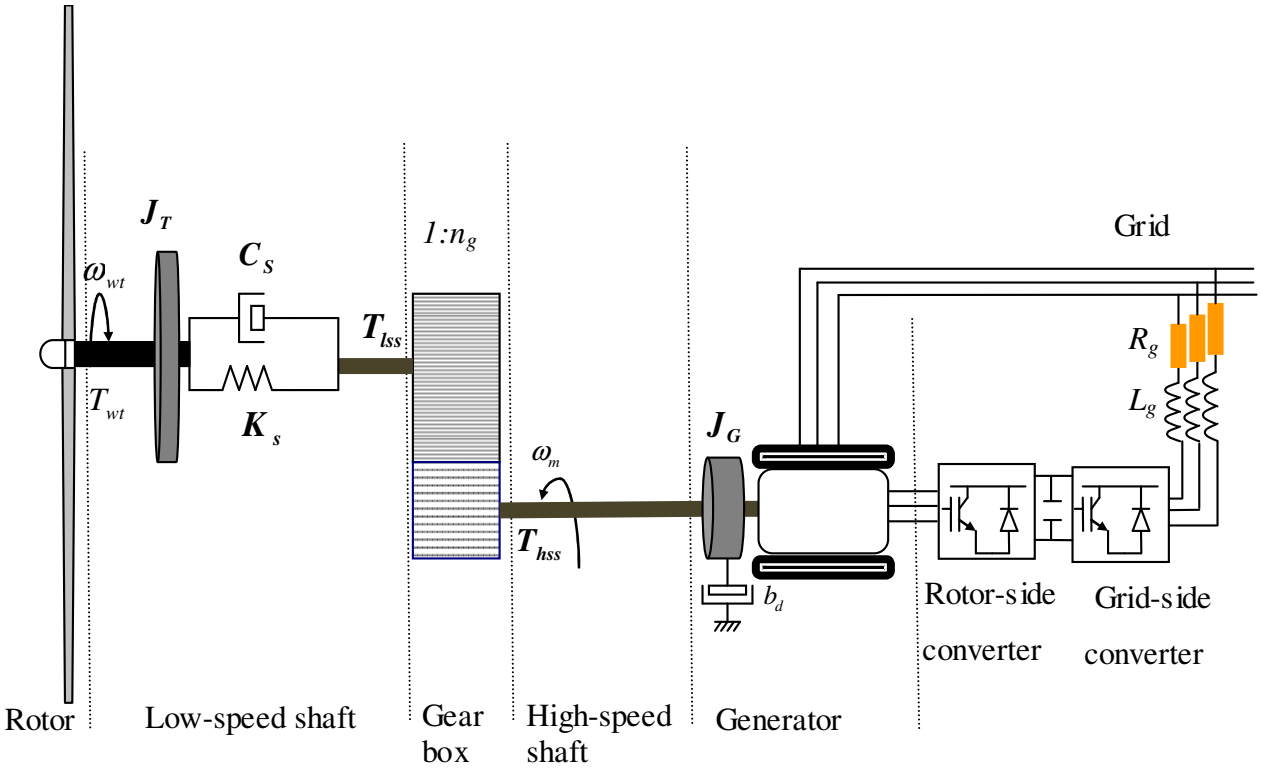


Figure 2.6: Wind turbine with a DFIG

2.4.1 Rotor torque model

When the wind speed is under-rated, the aim of the controller is to maximise the output power of the turbine. When losses in the drive-train are disregarded, equations (2.13) and (2.14) detail the aerodynamic torque T_{wt} at the turbine shaft. Parameters C_p and λ_i are estimated using equations (2.19) and (2.20). These equations were applied to predict and generate C_p curves for 5MW wind turbine [15].

$$C_p = c_1(c_2 - c_3\beta - c_4)e^{-c_5} + 0.0068\ \lambda \tag{2.19}$$

where:

C_1= 0.5176, C2=$\frac{116}{\lambda_i}$, C_3= 0.4, C_4= 5, C_5= $\frac{21}{\lambda_i}$ and λ_i is given by:

$$\lambda_i = \frac{(\lambda + 0.08\beta)(\beta^3 + 1)}{(\beta^3 + 1) - 0.035(\lambda + 0.08\beta)} \tag{2.20}$$

2.4.2 Two-mass drive-train model

When the dynamic model is compared with other mechanical models of a wind turbine, the first concern is the dynamic model of the drive-train since it is necessary to give priority to the parts of the dynamic structure of the wind turbine which contribute to grid integration [23, 24].

Often, a two-mass drive-train model is applied when analysing the interaction of the wind turbine with the grid. Therefore, the drive-train exerts a significant influence on the power fluctuations [24, 28]. In addition, the torque control can assist in dampening mechanical oscillations. Figure 2.7 shows a two-mass drive-train that can be described mathematically by equation (2.21); the corresponding Simulink model is illustrated in Figure B. 4.

$$\begin{bmatrix} \dot{\theta}_{wt} \\ \dot{\theta}_m \\ \\ \dot{\omega}_{wt} \\ \dot{\omega}_m \end{bmatrix} = \begin{bmatrix} \omega_{wt} \\ \omega_m \\ \\ \frac{1}{J_T}(T_{wt} - T_{lss}) \\ \frac{1}{JG}(T_{hss} - T_e - b_d\omega_m) \end{bmatrix} \tag{2.21}$$

where J_T is turbine inertia, J_G is generator inertia, $b_d \geq 0$ is the damping coefficient; and θ_{wt} and θ_m represent the turbine rotor angle and generator rotor angle, respectively.

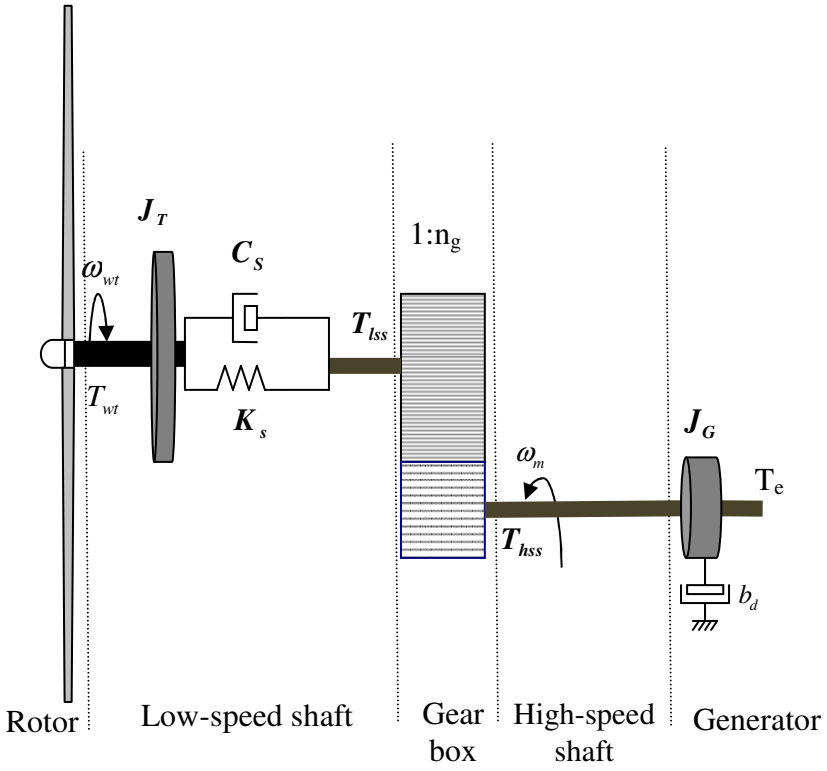


Figure 2.7: Two-mass drive-train model with gearbox [16]

In steady state the power balance is given by $T_{wt}\omega_{wt} = T_e\omega_m + b_d\omega_m^2$ or $T_{wt}\omega_{wt} = T_e\omega_m$ at $b_d=0$. The low-speed shaft torque can be written as [16]:

$$T_{lss} = K_s\theta_k + C_s\dot{\theta}_k = K_s(\theta_{wt} - \frac{\theta_m}{n_g}) + C_s(\omega_{wt}\frac{\omega_m}{n_g}) = n_g T_{hss} \tag{2.22}$$

where

$$\theta_k = \theta_{wt} - \frac{\theta_m}{n_g} \tag{2.23}$$

and K_s is the torsional stiffness of the low-speed shaft, C_s is the torsional damping of the low-speed shaft, n_g is the gearbox ratio, T_{lss} is the low-speed shaft torque and T_{hss} is the high-speed shaft torque.

2.4.3 Doubly-fed induction generator model

In order to analyse the dynamic modelling of the DFIG, the following assumptions are made [9, 30]:

- the DFIG is modelled using physical equations; and
- the control systems for the decoupled control of the active and reactive powers are represented.

The dynamic performance of AC generator is complex because the three-phase rotor windings move with respect to three-phase stator windings. The analysis can be simplified greatly by transforming the three-phase stator and rotor windings (with angular displacement) to the two-phase windings are called *d-q* frame (see Appendix D). Then the state equations of the 4th order DFIG model can be expressed in the *d-q* frame as [9, 33]:

$$\begin{bmatrix} \frac{d\psi_{ds}}{dt} \\ \frac{d\psi_{qs}}{dt} \\ \frac{d\psi_{dr}}{dt} \\ \frac{d\psi_{qr}}{dt} \end{bmatrix} = \begin{bmatrix} -R_s i_{ds} + \omega_s \psi_{qs} \\ -R_s i_{qs} - \omega_s \psi_{ds} \\ -R_r i_{dr} + \omega_{sl} \psi_{qr} \\ -R_r i_{qr} - \omega_{sl} \psi_{dr} \end{bmatrix} + \begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{dr} \\ v_{qr} \end{bmatrix} \tag{2.24}$$

where ψ_{ds} and ψ_{qs} are the stator flux *d-q* components and ψ_{dr} and ψ_{qr} are the rotor flux *d-q* components, i_{ds}, i_{qs}, i_{dr} and i_{qr} are the stator and rotor current *d-q* components, respectively. R_s and R_r are the stator and rotor resistances, $\omega_s = 2\pi f_g / n_p$ (rad/s) is the stator field frequency, n_p is the pole pairs number, f_g is the grid frequency; $v_{ds}, v_{qs}, v_{dr}, v_{qr}$ are the stator and rotor voltage *d-q* components, respectively. The stator and rotor flux ψ equations are given by:

$$\begin{bmatrix} \psi_{ds} \\ \psi_{qs} \\ \psi_{dr} \\ \psi_{qr} \end{bmatrix} = \begin{bmatrix} L_s i_{ds} + L_m i_{dr} \\ L_s i_{qs} + L_m i_{qr} \\ L_r i_{dr} + L_m i_{ds} \\ L_r i_{qr} + L_m i_{qs} \end{bmatrix} \tag{2.25}$$

where L_s and L_r are the stator and rotor inductances, respectively and L_m is the mutual inductance. The electrical torque is:

$$T_e = n_p \frac{L_m}{L_s}(\psi_{qs} i_{dr} - \psi_{ds} i_{qr}) \tag{2.26}$$

where ω_{sl} is the difference between a synchronous speed and generator rotor speed.

$$\omega_{sl} = \omega_s - \omega_m \tag{2.27}$$

Thus, the slip (S) of the generator can be rewritten as:

$$S = \frac{\omega_{sl}}{\omega_s} \tag{2.28}$$

2.4.4 Power flow

The total mechanical power P_a of the DFIG system (see Figure 2.8) is simply the sum of stator power P_s and rotor power P_r when the power converter is assumed to be lossless, i.e.

$$P_a = P_s + P_r \tag{2.29}$$

The active power from the rotor is proportional to the slip of the generator:

$$S = \frac{\omega_s - \omega_m}{\omega_s} \tag{2.30}$$

$$P_r = -S\, P_s \tag{2.31}$$

where ω_s is the synchronous speed and ω_m is the generator rotor speed. Substitute equation (2.31) into equation (2.29), to obtain:

$$P_a = (1 - S) P_s \tag{2.32}$$

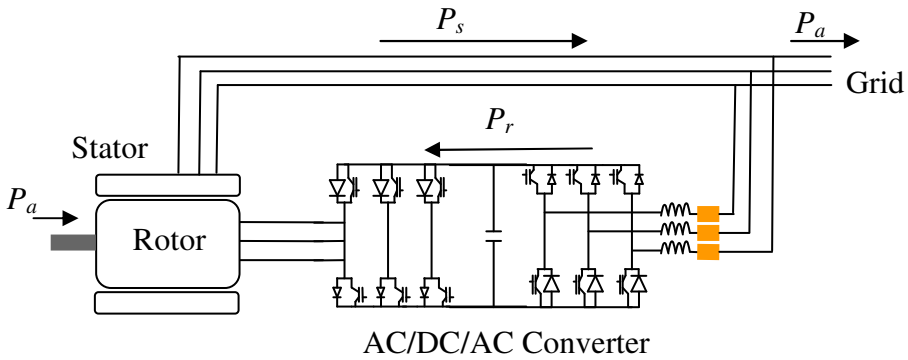


Figure 2.8: Power flow of a lossless DFIG wind turbine system

When operating in super-synchronous mode (the rotor speed above the synchronous speed), the generator feeds electrical power to the grid through both the rotor and the stator. If the generator is running in sub-synchronous mode, the electric power is delivered only into the rotor from the grid (traditional induction generators never produce power when running in sub-synchronous mode) [29].

If it is assumed that the converter is able to control the power flow at the converter-supply side at any point in time, then its reactive power is zero. This assumption is acceptable as a result of the converter rating being a maximum 30% of the generator rating and it is used primarily to supply the active power of the rotor to the grid. Therefore, the reactive power exchanged between the DFIG and the grid Q_{total} is equal to the reactive power in the stator Q_s (equation (2.33)).

$$Q_{total} = Q_s \tag{2.33}$$

The reactive power from the stator will be zero in the case of a strong power system or when there is no requirement for the DFIG to control the voltage. In this case, with its power factor close to unity, the DFIG supplies only active power and it is magnetised through the rotor. Under other conditions, the reactive power set point of the DFIG will be defined for the purpose of voltage control [30].

2.4.5 Grid-side converter modelling

Figure 2.9 illustrates the grid-side converter. The voltage across the inductors can be written as:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R_g \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + L_g \begin{bmatrix} \frac{di_a}{dt} \\ \frac{di_b}{dt} \\ \frac{di_c}{dt} \end{bmatrix} + \begin{bmatrix} v_a^c \\ v_b^c \\ v_c^c \end{bmatrix} \tag{2.34}$$

where $v_{a,}v_{b,}$ $v_{c,}$ $i_{a,}i_{b,}$ i_c are three-phase voltage and current signals, $v_a^c, v_{b,}^c v_c^c$ are three different control voltage outputs, R_g is the grid resistance and L_g is the grid inductance.

The three-phase signal with three quantities (v_a, v_b and v_c) can be transformed to *d-q* reference frame by using reference frame conversion (see appendix D). The voltage equations in the *d-q* frame are therefore given by:

$$v_d = R_g i_d + L_g \frac{di_d}{dt} - \omega_1 L_g i_q + v_d^c \tag{2.35}$$

$$v_q = R_g i_q + L_g \frac{di_q}{dt} + \omega_1 L_g i_d + v_q^c \tag{2.36}$$

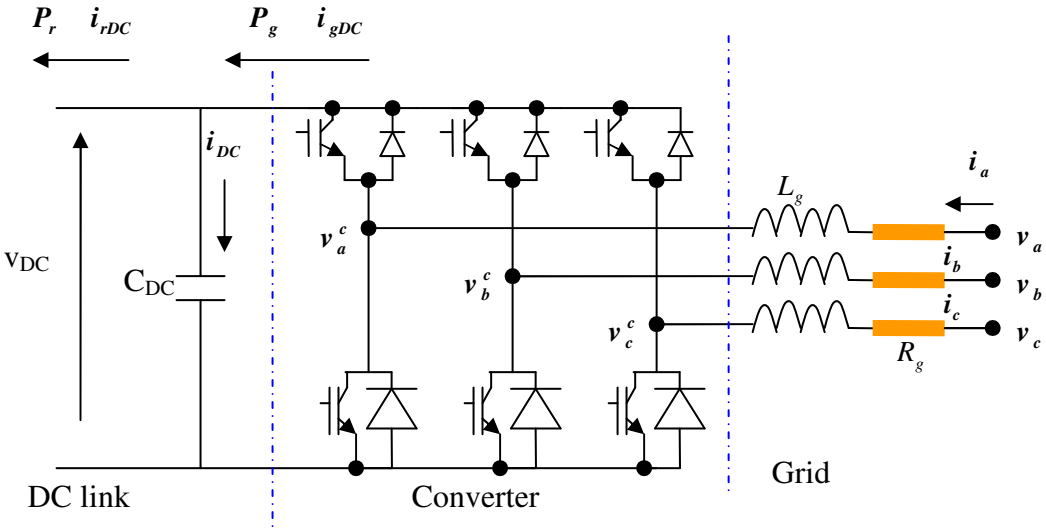


Figure 2.9: Schematic diagram of the grid-side converter [16]

Equations (2.35) and (2.36) can be converted to the state-space model and written as:

$$\dot{x} = \begin{bmatrix} -\frac{R_g}{L_g} & \omega_1 \\ -\omega_1 & -\frac{R_g}{L_g} \end{bmatrix} x + \begin{bmatrix} \frac{1}{L_g} & 0 \\ 0 & \frac{1}{L_g} \end{bmatrix} u_1 + \begin{bmatrix} -\frac{1}{L_g} & 0 \\ 0 & -\frac{1}{L_g} \end{bmatrix} u_2 \tag{2.37}$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \tag{2.38}$$

where:

State variables : $x = [i_d \quad i_q]^T$

Input : $u_1 = [v_d \quad v_q]^T$ and controller output $u_2 = [v_d^c \quad v_q^c]^T$

Output : $y = [i_d \quad i_q]^T$

Active and reactive grid powers can be calculated as:

$$P_g = v_d i_d + v_q i_q \tag{2.39}$$

$$Q_g = -v_d i_q + v_q i_d \tag{2.40}$$

2.4.6 DC-link model

It is assumed that the back-to-back converter is lossless and losses in the inductor resistance can be disregarded. Figure 2.10 shows an equivalent circuit of the DC-link model, where the power flows through the grid-side converter and the rotor-side converter. The power P_{DC} in the DC-link capacitor (C_{DC}) is dependent on the power delivered to the grid and the power delivered to the rotor circuit of the DFIG, as given by:

$$P_{DC} = v_{DC} i_{DC} = P_g - P_r \tag{2.41}$$

where v_{DC} and i_{DC} are DC-link voltage and current, respectively, P_g is grid power and P_r is rotor power.

The DC-link current is given by:

$$i_{DC} = C_{DC} \frac{dv_{DC}}{dt} \tag{2.42}$$

From equations (2.41) and (2.42), the DC-link voltage is:

$$\frac{dv_{DC}}{dt} = \frac{P_g - P_r}{v_{DC} C_{DC}} \tag{2.43}$$

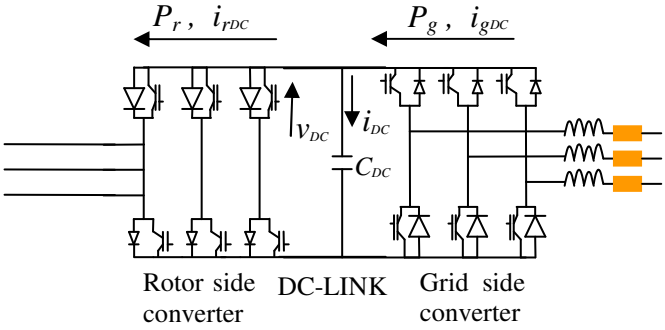


Figure 2.10: DC-link model (i_{rDC} and i_{gDC} are rotor and grid currents)

2.4.7 Summary

The total model of the wind turbine system will now be summarised.

- ***Aerodynamic Torque***

The torque acting on the rotor is:

$$T_{wt}(t) = \frac{\rho \pi R^2 v^3 Cp(\lambda(t), \beta(t))}{2\omega_{wt}}$$

- ***Drive-train***

The states for two-mass drive-train model are:

$$\dot{\theta}_{wt} = \omega_{wt}$$

$$\dot{\theta}_m = \omega_m$$

$$\theta_k = \theta_{wt} - \frac{\theta_m}{n_g}$$

$$\dot{\omega}_{wt} = \frac{1}{J_T}(T_{wt} - T_{lss})$$

$$\dot{\omega}_m = \frac{1}{JG}(T_{hss} - T_e - b_d \omega_m)$$

- ***Generator***

The states for the DFIG model are:

$$\frac{d\psi_{ds}}{dt} = -R_s i_{ds} + \omega_s \psi_{qs} + v_{ds}$$
$$\frac{d\psi_{qs}}{dt} = -R_s i_{ds} + \omega_s \psi_{qs} + v_{qs}$$
$$\frac{d\psi_{dr}}{dt} = -R_r i_{dr} + \omega_{sl} \psi_{qr} + v_{dr}$$
$$\frac{d\psi_{qr}}{dt} = -R_r i_{qr} - \omega_{sl} \psi_{dr} + v_{qr}$$

- ***DC-link model***

$$\frac{dv_{DC}}{dt} = \frac{P_g - P_r}{v_{DC} C_{DC}}$$

2.5 Fixed-point wind speed model

In order to model the wind speed it was assumed that, due to the roughness of the ground, turbulence occurred mainly from the friction between the air and the ground. Wind dynamics are influenced by the meteorological conditions and the local characteristics of a given site. Therefore, wind speed was modelled as a non-stationary random process which, as shown in equation (2.44), is generated by two components (see Figure 2.11) [1, 2, 3, 4].

$$v(t) = v_s(t) + v_t(t) \tag{2.44}$$

where:

$v_s(t)$: low frequency component

$v_t(t)$: turbulence component

A low-frequency component, $v_s(t)$, represents the very slow variations in wind speed, which represents the average wind speed of the site and gives an indication for available wind power. The low-frequency component can be modelled theoretically by using a Weibull distribution or a Rayleigh distribution [5].

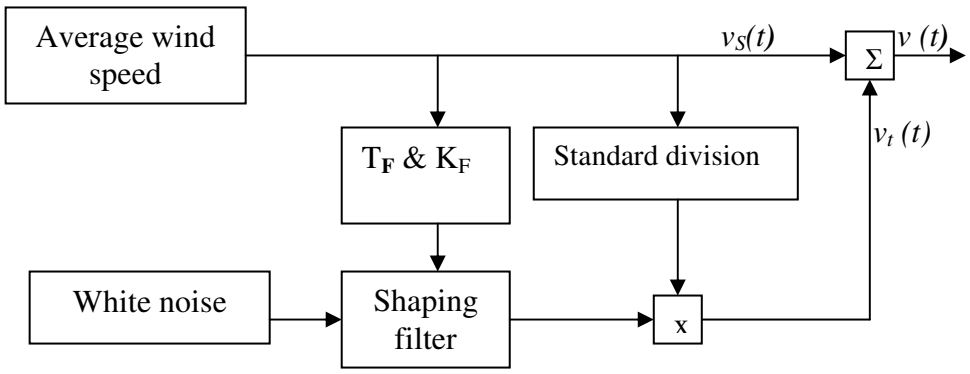


Figure 2.11: Non-stationary wind speed generation [9]. Simulink block diagram of the non stationary wind speed is presented in Figure B. 2

The turbulence component, $v_t(t)$, is due to the rapid variations in wind speed. It can be described mathematically as a zero mean normal distribution of which the standard deviation (σ_s) depends on the current value of the average wind speed (v_s), as given in equation (2.45). The intensity of the turbulence (I_t) is a measure of the global level of turbulence and depends on the roughness of the ground surface. This is defined in equation (2.46).

$$I_t = \frac{\sigma_s}{v_s} \tag{2.45}$$

$$I_t = \frac{1}{\ln(z / z_0)} \tag{2.46}$$

where z is the hub height of the wind turbine and z_0 is the 'roughness length' of the land surface.

The 'roughness length' (z_0) depends on the frontal area of the average element (facing the wind) divided by the ground width it occupies. A lower roughness length implies less exchange between the ground surface and the air, but also stronger wind near the ground. A terrain classification based on roughness length is given in Table 2.1.

A mathematical model of the turbulence's dynamical properties $v_t(t)$ can be developed by using either one of two approaches: Kaimal's spectrum is suitable for

experimental data, whereas von Karman's spectrum is more applicable to theoretically founded approaches [1]. By using von Karman's model, the transfer function of the shaping filter has the following form:

$$H_t(s) = K_F \frac{0.4T_F s + 1}{(T_F s + 1)(0.25T_F s + 1)} \tag{2.47}$$

The static gain K_F and time constant T_F depend on low-frequency wind speed (v_s). The time constant is calculated using equation (2.48) and the value of L_t is derived from practical experience. For instance, in the Danish standard, equation (2.50) is used to compute L_t. Equation (2.49) computes the static gain K_F [6].

$$T_F = L_t / \ v_s \tag{2.48}$$

$$K_F = \sqrt{4T_F} \tag{2.49}$$

$$L_t = \begin{cases} 150(m) & if\ z \geq 30m \\ 5z(m) & if\ z < 30m \end{cases} \tag{2.50}$$

This method was proposed by Welfonder [8] and adapted by Nichita [2]. It is implemented as shown in Figure B.2 in Appendix B to generate the non-stationary wind speed. For example, Figure 2.12 presents details of the wind speed profile when v_s=10 m/s.

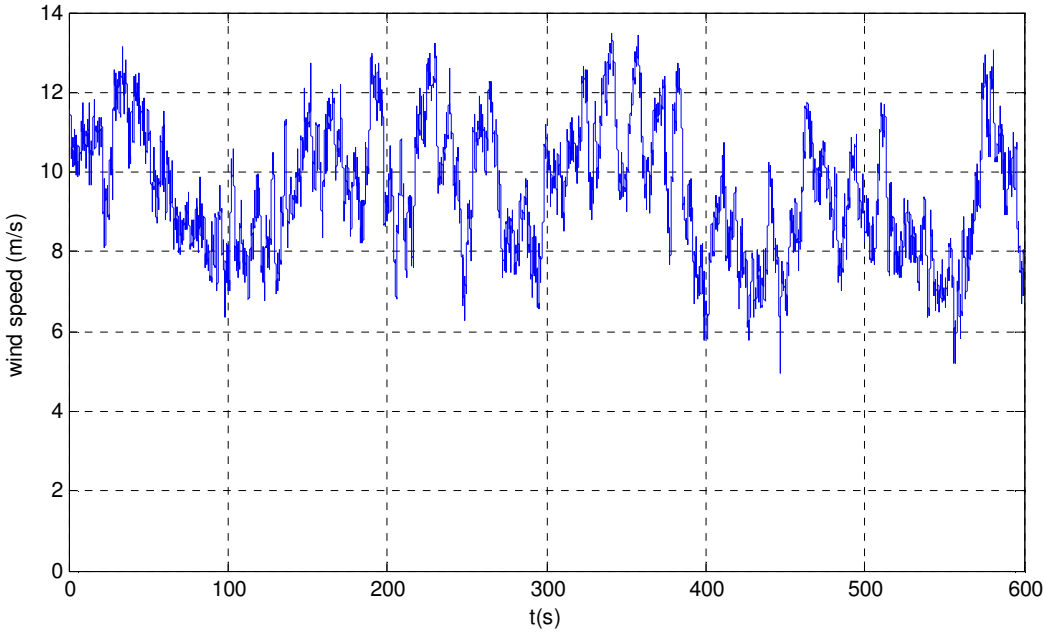


Figure 2.12: Wind speed profile for v_s=10 m/s

Table 2.1: Terrain roughness classification [7]

Class		Roughness length (m)	Landscape features
No.	Name		
1	sea	0.0002	open water, tidal flat, snow with fetch above 3 km
2	smooth	0.005	featureless land, ice
3	open	0.03	flat terrain with grass or very low vegetation, airport runway
4	roughly open	0.10	cultivated area, low crops, obstacles of height (H) separated by at least 20 H
5	rough	0.25	open landscape, scattered shelter belts, obstacles separated by 15 H or so
6	very rough	0.5	landscape with bushes, young dense forest etc separated by 10 H or so
7	closed	1.0	open spaces comparable with H, eg mature forest, low-rise built-up area
8	chaotic	> 2.0	irregular distribution of large elements, eg city centre, large forest with clearings

2.6 Wind turbine control

A wind turbine has many different types of vibration such as fore-aft vibrations and side-to-side vibrations of the tower. Moreover, each blade can vibrate in two main directions and the low-speed drive-train up to and including the gearbox has torsional vibrations [1, 19, 201]. The vibration frequencies depend on the turbine. It is not possible to suppress all vibrations because only the following two manipulation variables exist:

- the electrical torque through which the turbine speed can be controlled indirectly; and
- the pitch angle through which the aerodynamic torque can be regulated.

2.6.1 Torque control in the case of below-rated wind speed

Optimising the power output of the wind turbine below the rated wind speed is realised by using the torque control scheme for a variable-speed wind turbine As shown in equation (2.13), the reference electrical torque ($T_{e(ref)}$) can be written as:

$$T_{e(ref)} = \frac{P_{wt(\max)}}{\omega_{wt}} \tag{2.51}$$

From equations (2.13) and (2.14), $T_{e(ref)}$ at maximum output power is calculated as:

$$T_{e(ref)} = \frac{0.5\rho\pi R^2 C_{p(\max)} (\frac{R\omega_{wt}}{\lambda_{opt}})^3}{\omega_m} \tag{2.52}$$

where $C_{p(\max)}$ is the maximum power coefficient and λ_{opt} is the tip speed ratio at $C_{p(\max)}$.

The reference electrical torque can be rewritten as:

$$T_{e(ref)} = K_{opt}\omega_m^2 \tag{2.53}$$

where

$$K_{opt} = \frac{0.5\rho\pi R_w^5 C_{p(\max)}}{\lambda_{opt}^3 n_g^3} \tag{2.54}$$

It should be noted that in steady state $\omega_m = n_g \omega_{wt}$. Equation (2.53) is applied only when $\omega_m > 0$. In the infrequent cases when $\omega_m < 0$ (i.e. the turbine rotates backwards), it is better not to apply any electrical torque, i.e. $T_{e(ref)} = 0$, and instead to wait until the speed reverts to $\omega_m > 0$.

K_{opt} should be provided by the turbine manufacturer. However, K_{opt} might differ from turbine to turbine. Furthermore, it will change during a turbine's lifetime.

Figure 2.13 demonstrates the overall wind turbine system in the case when the wind speed is less than the rated wind speed. The block diagram for the wind turbine is implemented in Figure B.3 based on equations (2.13) and (2.19). Block diagrams for drive-train and generator are described in the preceding sections. Figure 2.14 gives the block diagram of the controller.

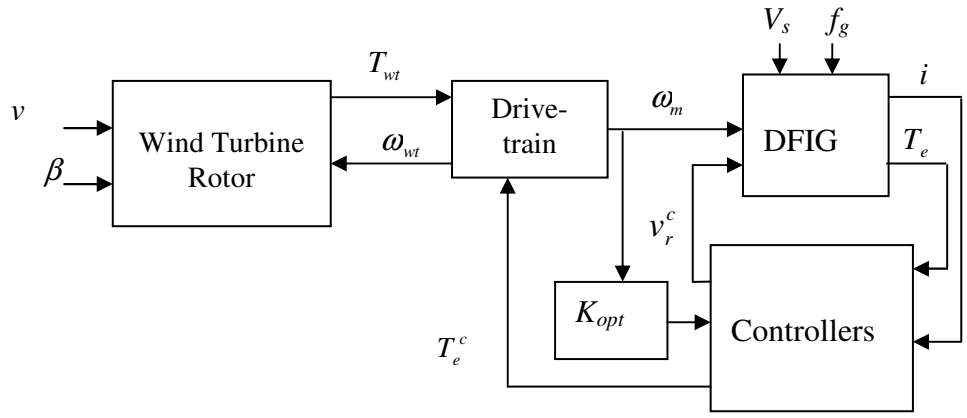


Figure 2.13: Block diagram for the control of the mechanical part of a wind turbine in the case of below-rated wind speed

Appendix B gives the Simulink diagrams of the turbine rotor, drive-train, generator and controller for the wind turbine system that is implemented in Matlab/Simulink. Figure B. 6 illustrates the controller in Figure B.1. It consists of a Simulink model to calculate the reference electrical torque in Figure B.7; a Simulink model of the reference rotor current calculation in Figure B.8; and a Simulink model of the rotor

voltage controller in Figure B.9. Here, PI controllers were used to assess the performance of the wind turbine model. As shown in Figure 2.14, the controller has electrical and mechanical inputs and outputs. Electrical inputs are stator and rotor currents of the generator ($i_{ds}, i_{qs}, i_{dr}, i_{qr}$) and the mechanical inputs are the torque and rotational speed of the generator. Electrical output control signals are the rotor voltage (v_{dr}^{c}, v_{qr}^{c}) and the mechanical output is the generator torque (T_e^c). The function of the controller is to control electrical torque and rotor voltage in order to obtain optimal output power. The tuning of proportional-integral (PI) controllers will be discussed in the next chapter.

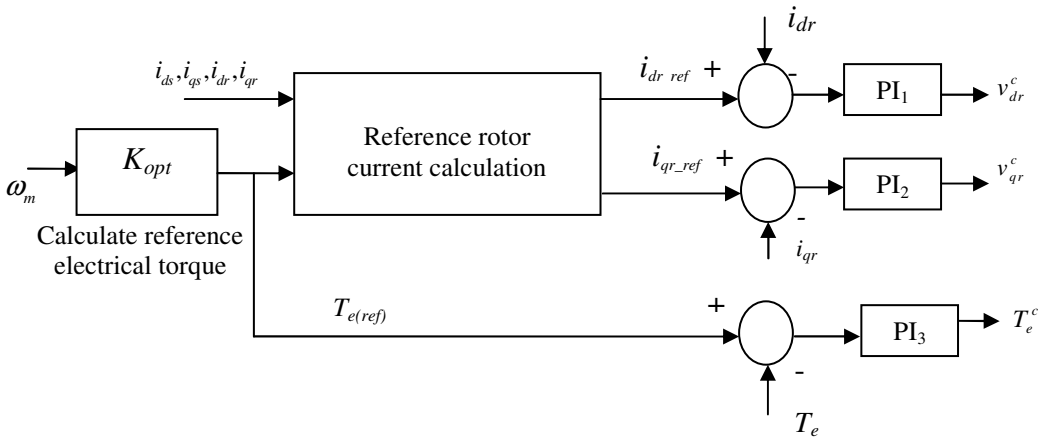


Figure 2.14: Block diagram of a controller

The electrical torque of the generator should be kept constant at a desired value. However, when dampened lightly, strong variations in wind speed might lead to oscillations in the drive-train, In addition, oscillations can damage the gearbox, which is one of the most expensive components in a wind turbine. Hence, there is a requirement to reduce these oscillations; one successful method is to control the generator torque, as shown in Figure 2.14.

In order to achieve a decoupled control between the stator active and reactive powers, the DFIG's *d-q* reference frame was used with the d-axis oriented along the

stator-flux vector position. Since the stator is connected to the grid, it is valid to make the following assumptions:

- The stator magnetising current space is the phasor $\vec{i}_{ms} = \left|\vec{i}_{ms}\right|$. In the steady state, it is $\left|\vec{i}_{ms}\right|$=constant.

- The frequency of the power supply to the stator is constant (f_g =50Hz) and, consequently, $\omega_s = \dfrac{2\pi\ f_g}{n_p}$.

- The stator resistance R_s can be disregarded (usually acceptable where a generator power is very high).

Then, from stator and rotor flux ψ, equation (2.25) can be rewritten as [36]:

$$\psi_{ds} = L_m \left|\vec{i}_{ms}\right|$$
$$\psi_{qs} = L_s i_{qs} + L_m i_{qr} = 0$$
$$\psi_{dr} = L_m^2 \left|\vec{i}_{ms}\right| + \sigma L_r i_{dr}$$
$$\psi_{qr} = \sigma L_r i_{qr}$$

where $\left|\vec{i}_{ms}\right| = \dfrac{\left|\vec{v}_s\right|}{\omega_s L_m}$, leakage coefficient $\sigma = 1 - \dfrac{L_m^2}{L_s L_r}$ and $\left|\vec{v}_s\right| = \sqrt{3}\ V_s$ (V_s is the stator voltage).

The state equations of the 4th order DFIG model (2.24), after some substitutions, can be written as:

$$
\begin{aligned}
v_{ds} &= R_s i_{ds} - \omega_s \psi_{qs} + \frac{d\psi_{ds}}{dt} \approx 0 \\
v_{qs} &= R_s i_{qs} + \omega_s \psi_{ds} + \frac{d\psi_{qs}}{dt} \approx \omega_s \psi_{ds} = |\vec{v}_s| \\
v_{dr} &= \sigma L_r \frac{di_{dr}}{dt} + R_r i_{dr} - (\omega_s - \omega_m)\sigma L_r i_{qr} \\
v_{qr} &= \sigma L_r \frac{di_{qr}}{dt} + R_r i_{qr} + (\omega_s - \omega_m)(\sigma L_r i_{dr} + \frac{\sqrt{3} L_m V_s}{\omega_s L_s})
\end{aligned}
\tag{2.55}
$$

From equation (2.26), the reference rotor current can be calculated as:

$$
i_{qr(ref)} = -\frac{L_s \omega_s}{n_p L_m V_s \sqrt{3}} T_{e(ref)} \tag{2.56}
$$

The rotor voltages v_{dr} and v_{qr} output from the controller are used to control the rotor voltages of the generator. i_{dr_ref} is given by the outer loop controller for grid integration. In this simulation, i_{dr_ref} is assumed to equal zero; this assumption is true when the difference between DC-link voltages V_{dc_ref} and V_{dc} equals zero.

The model was verified by generating a step response at v=10 m/s using the Simulink model of Figure B. 3. For wind turbine torque, Figure 2.15 illustrates that the response is slow, which is a result of the induction lag. This is due to an aerodynamic effect when the wind speed or the pitch angle changes.

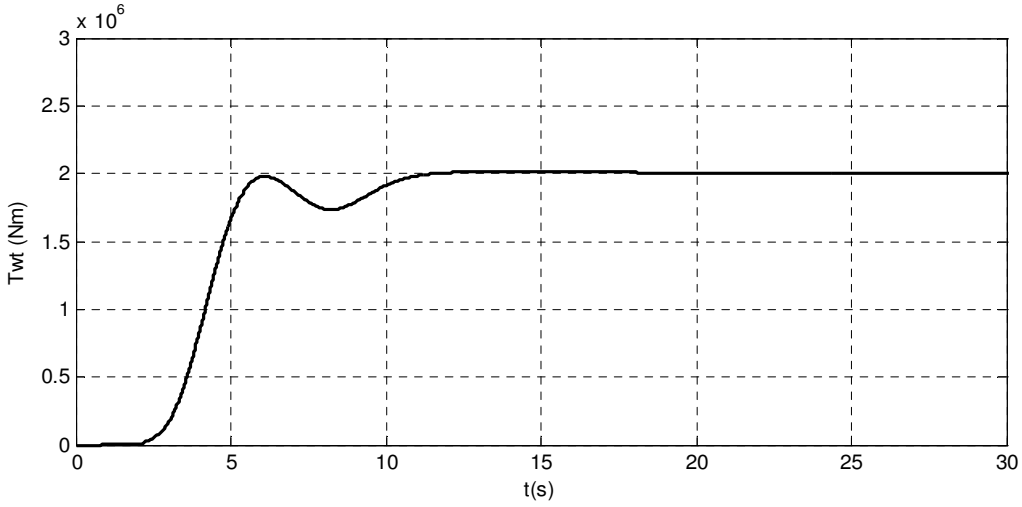


Figure 2.15: Wind turbine torque T_{wt} at wind speed = 10 m/s

2.6.2. Rotor torque control in the case of above-rated wind speed

In the case of wind speed above the rated wind speed, the main task is to maintain a generated power at the rated value. This can be realised by using the rotor current controller and the rotor-side converter to maintain a constant electrical torque and, also, by employing the pitch control to regulate the turbine speed (see Figure 2.16).

This section aims to evaluate the validity of the wind turbine model. Therefore, this chapter will not discuss the design of a controller for a non-linear model and pitch controller (those are the objectives of the next chapter).

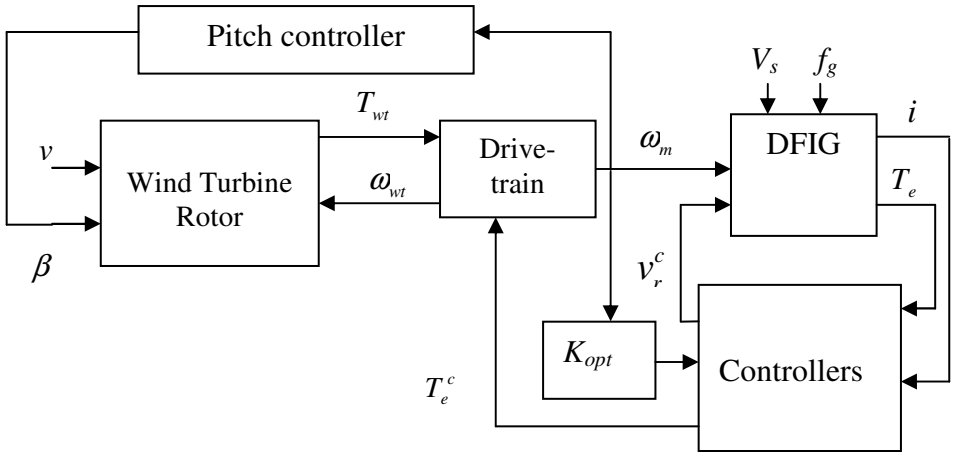


Figure 2.16: Block diagram for the control of the mechanical part of a wind turbine in strong wind

2.7 Simulation results

The simulation is implemented in three steps. Firstly, the response of the open loop of the wind turbine rotor was tested using a Simulink block diagram model of the wind turbine rotor, as shown in Figure B.3. After that, the performance of the wind turbine system was examined in cases of constant wind speed and non-stationary wind speed using a Simulink model for all subsystems of the wind turbine (see Figure B.1).

I. Testing the open loop of the wind turbine rotor model

Figures 2.17 and 2.18 show the performance of the power coefficient *vs.* the tip speed ratio and of the torque coefficient *vs.* the tip speed ratio at different pitch angles. Figure 2.19 illustrates the relationship between the power output of the rotor and the wind turbine rotational speed at different wind speeds (v =5, 10, 15 m/s) and constant pitch angle equal zero. The theoretical maximum efficiency of a wind turbine is given by the Betz Limit, and is around 59 percent. Practically, wind turbines operate below the Betz Limit. For example, in Figure 2.17 where the pitch angle= 0^o, it is operated at the optimal tip speed ratio of 8.2; its power coefficient is around 0.47. This suggests that for maximum power extraction, a wind turbine should be operated around these values when β=0 and a wind speed is less than the rated wind speed.

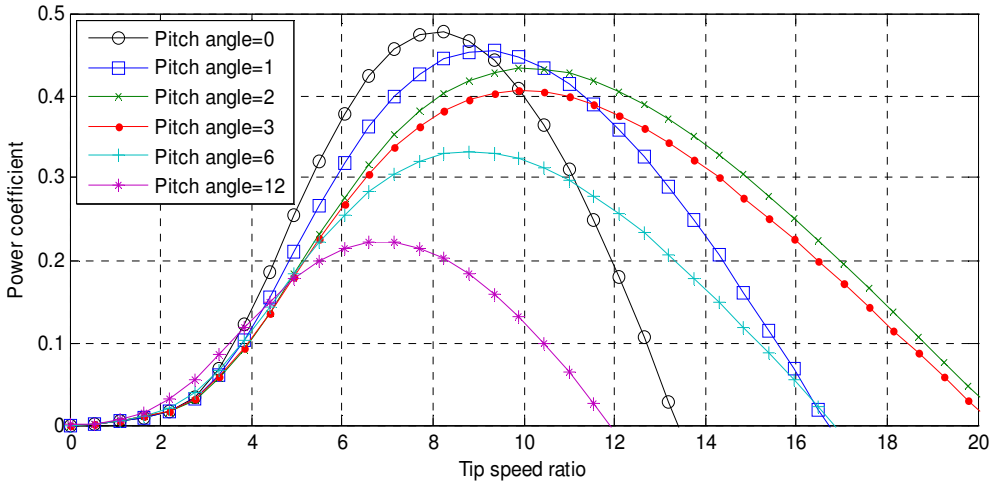


Figure 2.17: C_p curves for a 5MW wind turbine for different pitch angles (pitch=0^o, 1^o, … 12^o)

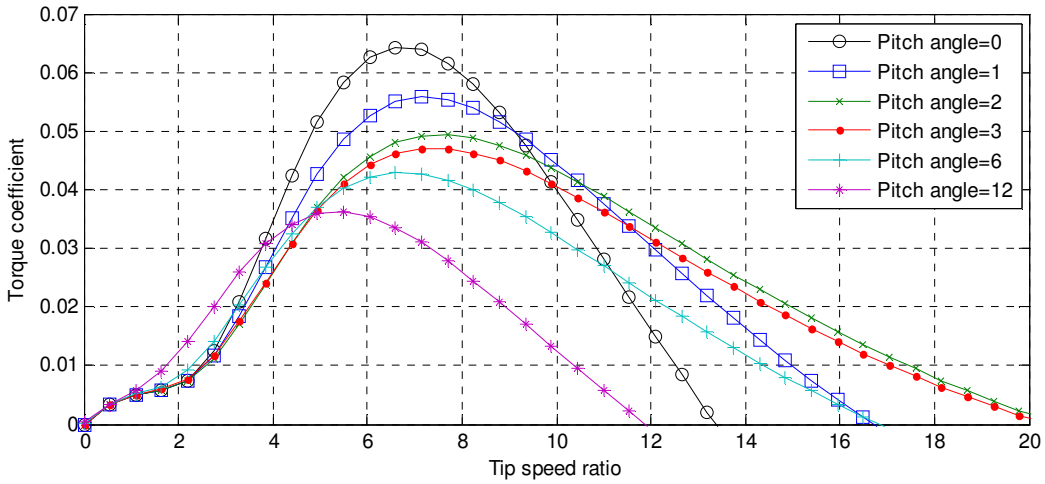


Figure 2.18: C_t curves for a 5MW wind turbine for different pitch angles (pitch=0^o, 1^o, … 12^o)

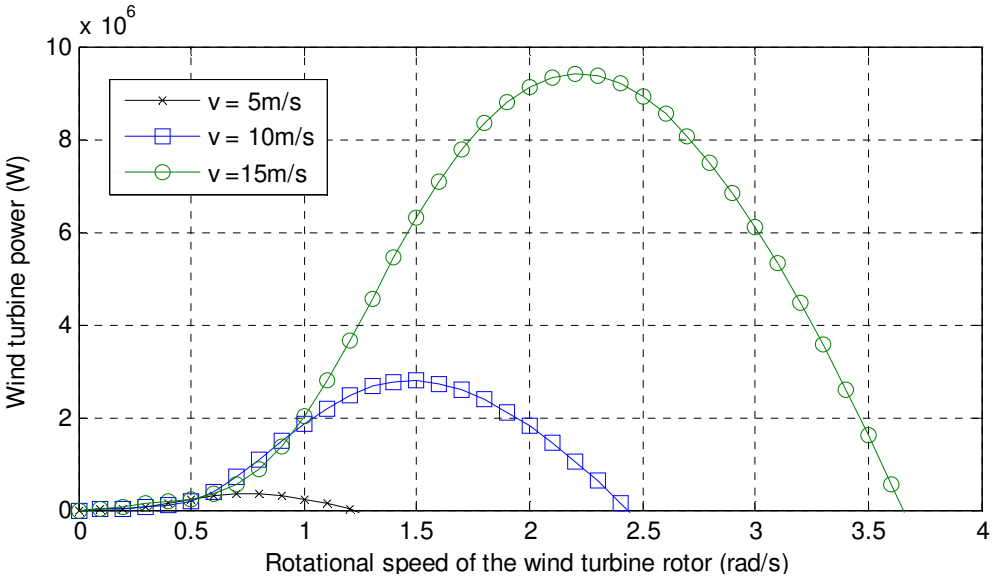


Figure 2.19: 5MW power characteristics, for pitch angle = 0

II. Test model for all subsystems of the wind turbine at constant wind speed

In this case, the results were based on a constant wind speed of 10 m/s and pitch angle =0^{o}. Consequently, as shown in Figures B.6 and B. 9, the gains of PI controllers were defined using Matlab ($K_{p1}, K_{p2}, K_{p3,} K_{i1}, K_{i2}, K_{i3}$ =60, 0.9, 5, 0.1, 0.001 and 2, respectively).

Figures 2.15 and 2.20 show that, for wind speed v =10 m/s, the maximum wind turbine rotor speed is $\omega_{wt} = 1.4$ rad/s and, at that speed, the maximum wind turbine torque T_{wt} = 2000 KNm is achieved. The corresponding maximum power is P_{wt} =2.77MW, as shown in Figures 2.19 and 2.21.

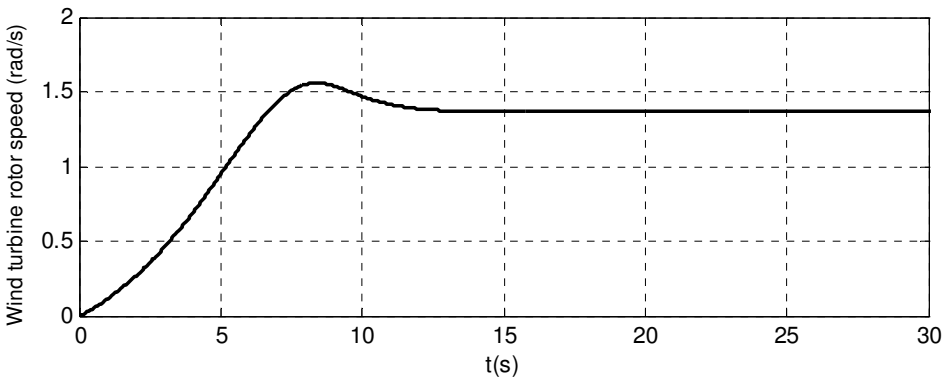


Figure 2.20: Wind turbine rotor speed at wind speed = 10 m/s

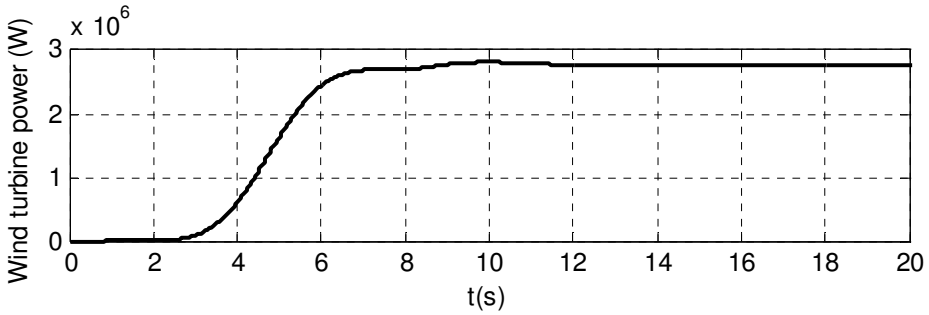


Figure 2.21: Wind turbine power at wind speed = 10 m/s

The value of power coefficient C_p is maintained at its maximal value in the steady state (see Figure 2.22). This demonstrates the achievement of the control objective to capture the maximum wind energy with maximum C_p = 0.47 and optimal λ = 8.2 (see Figure 2.17). Moreover, Figure 2.18 shows that the torque coefficient C_t = 0.057 at the optimal tip speed ratio is equal to the calculated value $C_t = C_{p\max} / \lambda_{opt}$.

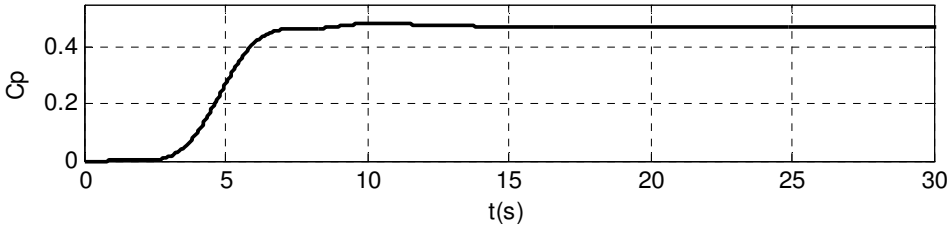


Figure 2.22: Plot of Cp at wind speed = 10 m/s, pitch angle= 0^o

III. Test model for all subsystems of the wind turbine at stationary wind speed

Here, the third case of simulation at a more realistic profile of wind speed was implemented (see Figure 2.12). As described in Table 2.1, the turbulence intensity was calculated for a wind turbine in the sea. In Figure B.2, the generated wind speed along the turbine axis was based on the Simulink block diagram of the non-stationary wind speed.

Simulation results are shown in Figures 2.23, 2.24 and 2.25 for the generator rotor speed, wind turbine torque and electric torque, respectively. The figures illustrate the

response oscillation around their optimal value that is expected to happen when the controller is designed as a constant operating point, so that the optimum torque and maximum power are achieved.

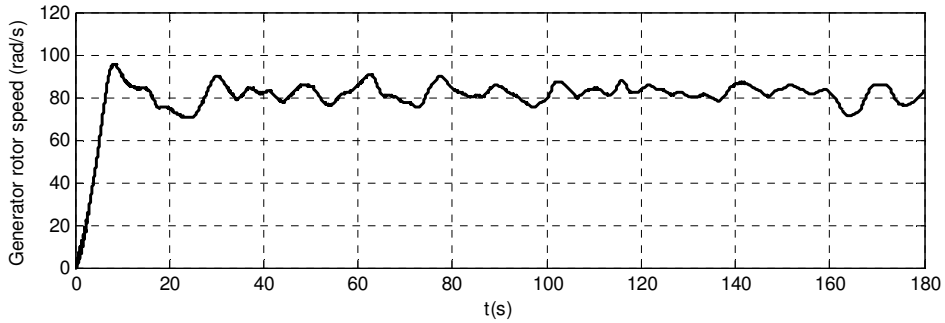


Figure 2.23: Generator rotor speed, in the case of stationary wind speed (average wind speed of 10 m/s)

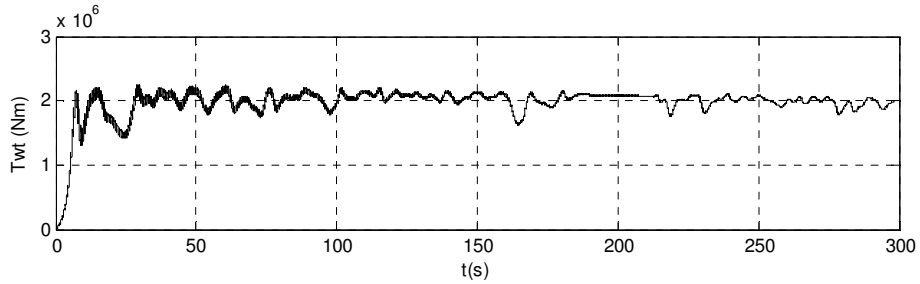


Figure 2.24: Wind turbine torque, in the case of stationary wind speed (average wind speed of 10 m/s)

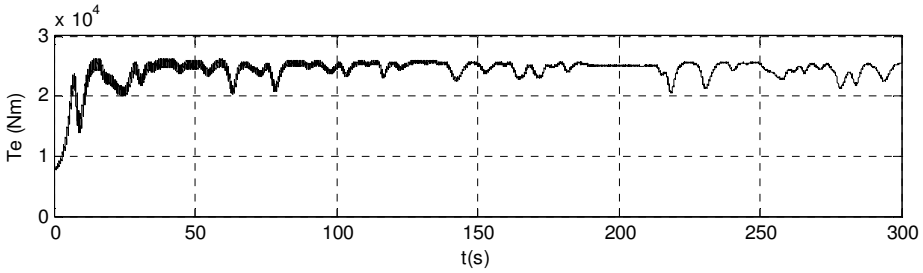


Figure 2.25: Electric torque, in the case of stationary wind speed (average wind speed of 10 m/s)

2.8 Conclusions

In this chapter, the mathematical models for wind speed and the main components, particularly aerodynamic rotor, two mass drive-train, generator, electrical torque controller and DFIG rotor current controllers were presented within a 5MW wind turbine system. Matlab/Simulink was used to implement all these models. The simulation results demonstrated that the physical wind turbine system can be modelled by using Matlab based on a mathematical model. Accordingly, Matlab is a very powerful tool in developing models and monitoring methods. Nevertheless, for a wind turbine system, the interconnections between the subsystems were complex.

An advanced control method should be used for a non-linear system based on a realistic wind-speed profile. Several researchers have investigated this issue and numerous strategies are being used to reduce the deviations in the characteristics surrounding the optimum operating points of the various regimes [9].

3 PI Controller Design for Wind Turbine

In this chapter, PI controllers for controlling electrical torque and pitch angle were designed. Two methods were proposed to calculate the gains of a PI pitch angle controller for a non-linear model of a 5MW wind turbine; the first method was analytical and the second was based on simulation. Firstly, the power coefficient characteristics were calculated for different pitch angles. Secondly, the output powers *vs.* rotor speed curves were simulated from cut-in to cut-out wind speeds. The results from the first and second analyses were used to find the control gains at different wind speeds. Finally, a wind-turbine model was used to determine the tracking characteristics of the turbine.

In order to design an electrical torque controller, an IMC-based PI was used to determine the gains of the current and torque controller which achieved good static and dynamic performance.

3.1 Introduction

Pitch variable-speed wind turbines have become the dominating type of yearly installed wind turbines in recent years. Typically, there are two control strategies for variable-speed wind turbines. At low wind speed, below a rated value, the speed controller can adjust continually the rotor speed to maintain the speed at a level which gives the maximum power coefficient. Then, for a variable-speed wind turbine [17], the turbine's efficiency will be increased by using the torque control scheme. In the case of wind speed above the rated wind speed, the main task is to maintain the generated power at the rated value. This can be realised by using a pitch angle control to regulate the turbine speed. Small changes in pitch angle can affect the power output. The purpose of the control can be summarised in the following three aims:

- Optimising the power output when a wind speed is less than the rated wind speed (see Figure 2.13 in Chapter 2).
- Keeping the rotor power within the design limits when the wind speed is above the rated wind speed (see Figure 2.16 in Chapter 2).
- Minimising the fatigue loads of the turbine's mechanical components.

The design of the controller must take into account the effects of the loads, and the controller should ensure that any control action would not result in extreme loads. It is possible to go further than this by, as an additional objective, designing the controller clearly to reduce certain fatigue loads.

In an active PID pitch controller, the sensitivity of the aerodynamic power to the rotor-collective blade-pitch angle is negative when the wind speed is above the rated wind speed. Then, with positive control gains, the derivative term will increase the effective inertia of the drive-train. Hansen et al. (2005) [23] recommended using a PI controller. However, based on test results, Boukhezzar et al. (2007) [38] suggested using only a proportional pitch controller which showed that a more complex controller (PI and PID) would make the pitch control more turbulent without any significant improvement in power regulation performance. Moreover, it was shown that using an advanced control strategy, such as a linear quadratic Gaussian control design technique, ensured better power tracking than the PID. However, this turned out to still be insufficient to meet all the control objectives [39].

3.2 Pitch Controller

As shown in Figure 3.1, adjusting the pitch angle to the blades provided an effective means of limiting turbine performance in strong wind speeds. We used electric or hydraulic pitch servos to put the blades into the desired pitch angle.

The blade control strategies can be classified according to the input signals used to generate the pitch set point [40] as follows:

- Ideally, the pitch angle reference can be obtained from the curve at the pitch angle versus wind speed. This control strategy is not an acceptable method since the effective wind speed cannot be measured accurately.
- The error signal of the generated power is sent to a controller to produce a reference pitch angle.
- The error between the generator rotor speed and its set point is sent to the controller to produce a reference value to the pitch angle. This method is the most popular since it is the most accurate.

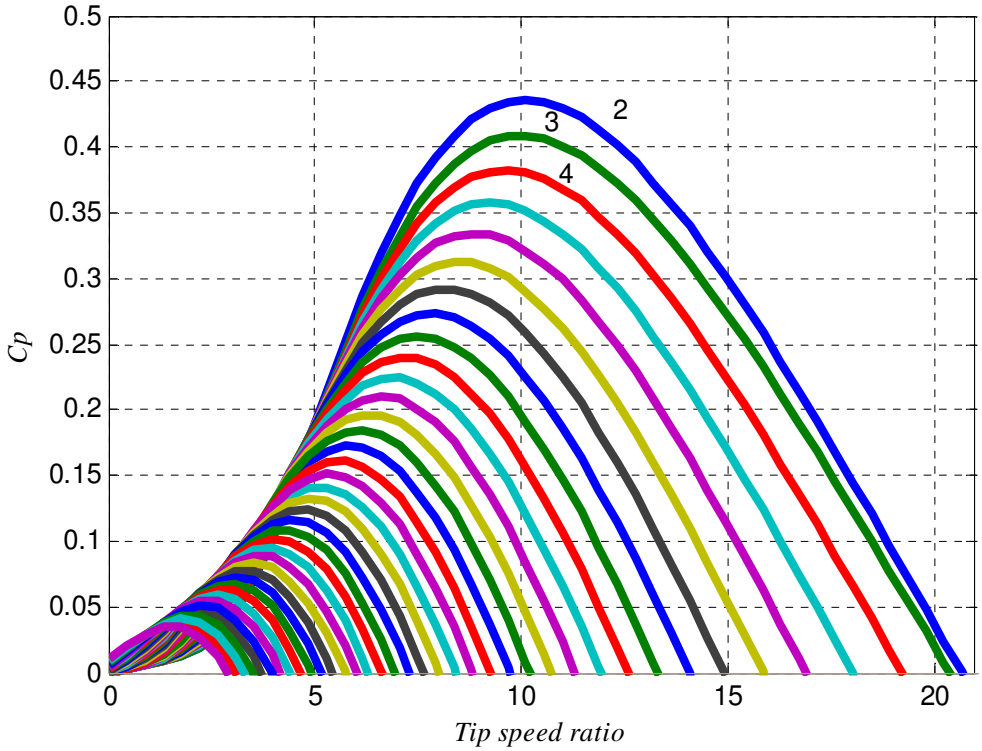


Figure 3.1: Power coefficient characteristics of 5MW wind turbine for different pitch angles (pitch=2°, 3°, 4°… 39°)

The power in the wind, P, is proportional to the cube of the wind speed and can be expressed as:

$$P = 0.5\rho A v^3 \tag{3.1}$$

where ρ is the air density, A is the area swept by the blades and v is the wind speed.

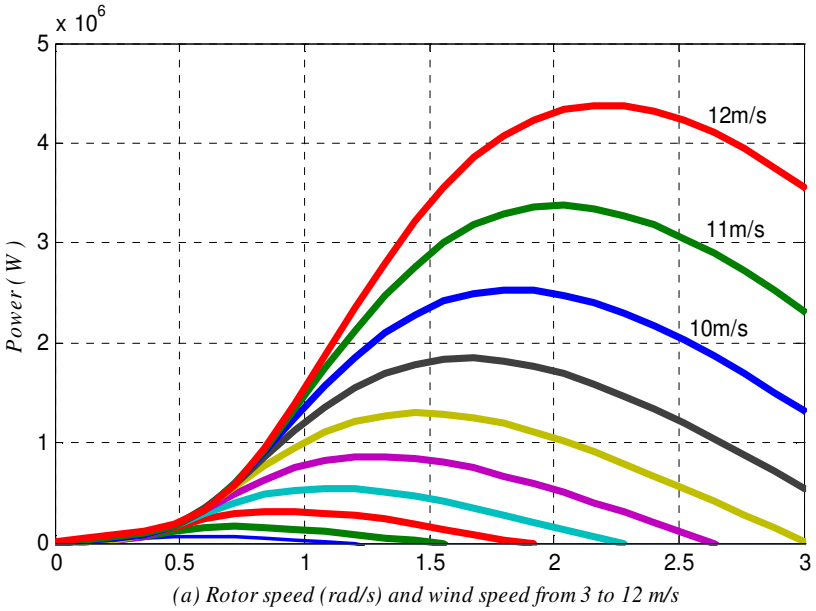
A wind turbine can only extract part of the wind power which is limited by the Betz limit (maximum 59%). This fraction is described by the power coefficient of the turbine which is a function of the blade-pitch angle and the tip speed ratio. Therefore, the mechanical power of the wind turbine extracted from the wind is:

$$P_{wt} = 0.5\,\rho A v^3\, C_p(\beta, \lambda) \tag{3.2}$$

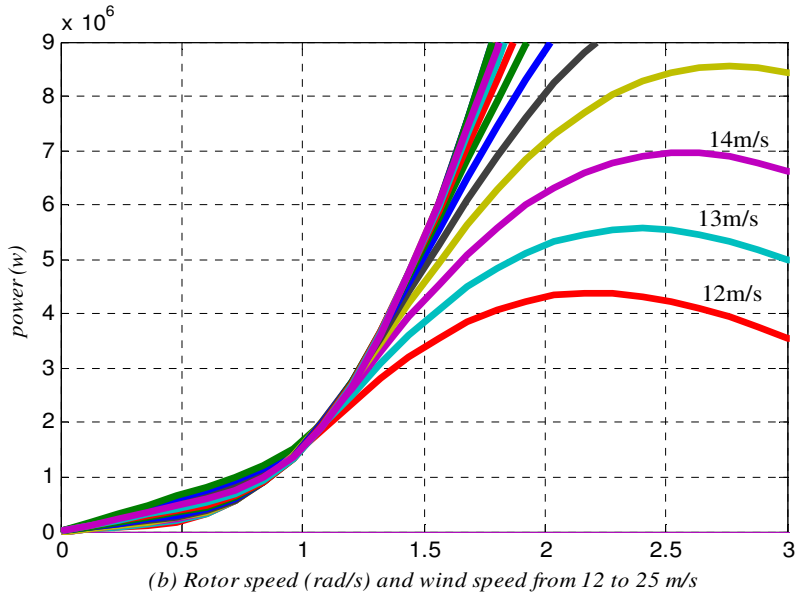
where C_p is the power coefficient of the wind turbine, β is the blade-pitch angle and λ is the tip speed ratio. The value of $C_p(\beta, \lambda)$ is highly nonlinear and varies with the wind speed, the rotational speed of the turbine and turbine blade parameters, such as the pitch angle. Here, equations (2.19) and (2.20) in Chapter 2 were used to calculate the value of the $C_p(\beta, \lambda)$.

The tip speed ratio is the ratio between the blade tip speed and the wind speed. Therefore, any variance in the rotor speed or the wind speed induces a change in the tip speed ratio, leading to power coefficient variation.

As shown in Figure 3.2, the power extracted by the turbine increases as the wind speed increases at the rated wind speed and the generating power reaches the rated power of the turbine. If the wind speed continues to rise, the output power will increase, also, and, consequently, the control system is required to keep the power constant at the design limit. For safety considerations, the turbine must shut down at speeds exceeding the cut-out wind speed.



(a) Rotor speed (rad/s) and wind speed from 3 to 12 m/s



(b) Rotor speed (rad/s) and wind speed from 12 to 25 m/s

Figure 3.2: Output power from wind turbine rotro *vs.* rotor speed for different wind speeds (3, 5, and 6…25 m /s) for

3.3 Actuator model

The pitch system consists of three identical pitch actuators; each has a controller and a hydraulic actuator, which is used to turn the blade along their longitudinal axis (see Figure 3.3). Therefore, the section describes only one actuator model. The actuator

model describes the dynamic behaviour between a pitch demand β_d from the pitch controller and the measurement of a pitch angle β .

The dynamics of the blades are nonlinear with saturation limits on both the pitch angle and the pitch rate. In a closed loop, the pitch actuator can be modelled as a first-order dynamic system with saturation in the amplitude and derivative of the output signal [39]. Figure 3.4 shows a block diagram of the first-order actuator model.

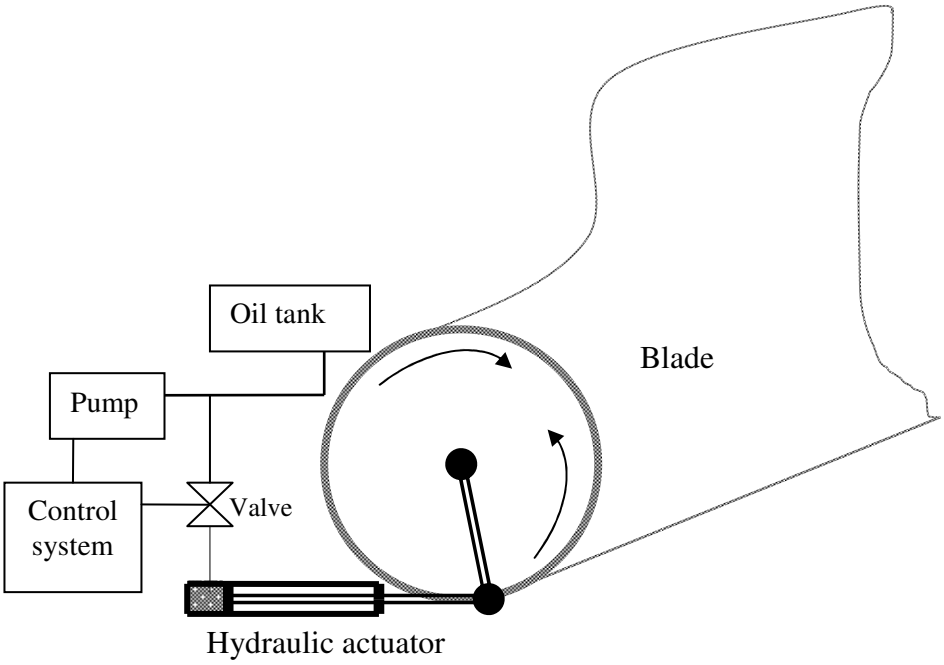


Figure 3.3: Hydraulic pitch system

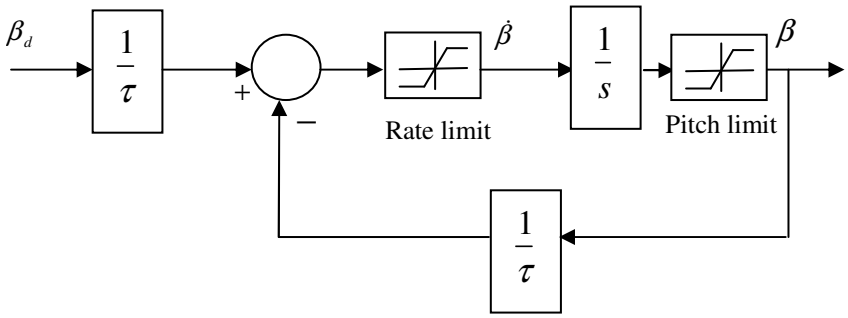


Figure 3.4: Model of the pitch angle actuator

The dynamic behaviour of the pitch actuator operating in its linear region is described by the differential equation:

$$\dot{\beta} = -\frac{1}{\tau_\beta}\beta + \frac{1}{\tau_\beta}\beta_d \tag{3.3}$$

From the above equation, the transfer function for the actuator is:

$$\frac{\beta}{\beta_d} = \frac{1}{\tau_\beta s + 1} \tag{3.4}$$

where τ_β is a time constant of the pitch actuator. Typically, pitch angle ranges from 0^o to 30^o and varies at a maximum rate of 10^o/s.

3.4 Conventional pitch angle control methods

As described below, the methodology of conventional pitch angle controls is divided into collective and individual blade-pitch controls.

3.4.1 Collective blade-pitch control

3.4.1.1 Blade-pitch control system using a simple single-degree-of-freedom

In order to design a blade-pitch control system using a simple single-degree-of-freedom (see Figure 3.5), Jonkman et al. (2009) [41] presented in the following equation the operating point of the blade-pitch angles' perturbation about their operating point as:

$$\Delta\beta = n_g\left(K_p\Delta\omega + K_I\int_0^t \Delta\omega dt + K_D\Delta\dot{\omega}\right) \tag{3.5}$$

where K_P, K_I, and K_D are the blade-pitch controller proportional, integral, and derivative gains, respectively.

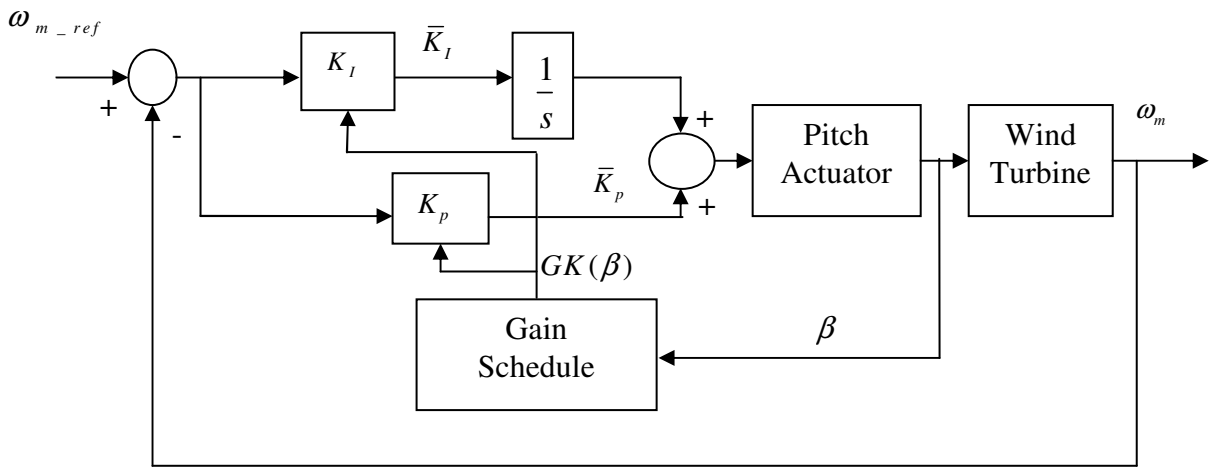


Figure 3.5: Simple flowchart of the blade-pitch control system

In an active pitch controller, the sensitivity of aerodynamic power to the rotor-collective blade-pitch angle, $\partial P / \partial \beta$, is negative when the wind speed is above the rated wind speed (see Figure 3.6).

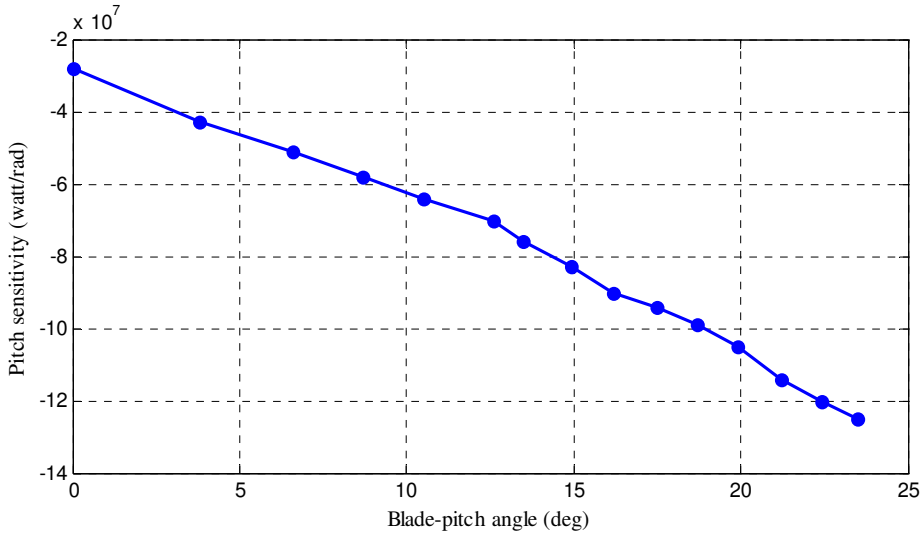


Figure 3.6: The sensitivity of aerodynamic power ($\partial P / \partial \beta$) to rotor-collective blade-pitch angle (above rated wind speed) [23].

Then, with positive control gains, the derivative term acts to increase the effective inertia of the drive-train, the proportional term adds damping, and the integral term adds restoring. Additionally, for the reason that the generator torque drops with increasing speed error (to maintain constant power) in Region 3, the generator-torque controller introduces a negative damping in the speed error response [41]. This

negative damping must be compensated by the proportional term in the blade-pitch controller. Hansen et al. (2005) [23] recommended neglecting the derivative gain, ignoring the negative damping from the generator-torque controller and aiming for the response characteristics given by natural frequency (ω_n) = 0.6 rad/s and damping ratio (ξ)= 0.6 to 0.7. This specification leads to direct expressions for choosing PI gains once the sensitivity of aerodynamic power to rotor-collective blade pitch, *dP/dβ*, is known:

$$K_p = \frac{2J\ \omega_{wt\_ref}\xi\omega_n}{n_g}\left[-\frac{\partial P}{\partial \beta}\right]^{-1} \tag{3.6}$$

$$K_I = \frac{J\ \omega_{wt\_ref}\omega_n^2}{n_g}\left[-\frac{\partial P}{\partial \beta}\right]^{-1} \tag{3.7}$$

where ω_{wt_ref} is the rated low-speed shaft rotational speed, and *J* is the drive-train inertia. The blade-pitch sensitivity, $\partial P / \partial \beta$, is an aerodynamic property of the rotor which depends on the wind speed, the rotor speed and the blade-pitch angle.

The corresponding values of the constants K_p and K_I of the PI-controller can be calculated when the pitch sensitivity $\partial P / \partial \beta$ is known (see Figure 3.6). This quantity must be calculated from the aerodynamic properties of the rotor. The calculated pitch sensitivity gives a large variation with wind speed (numerically increasing with increasing wind), and, consequently, constant values of K_P and K_I will not give the desired result. The plot of the pitch sensitivity as a function of the pitch angle in Figure 3.6 shows a nearly linear relation [41]. This implies that a gain correction factor, $GK(\beta)$, multiplied with both K_P and K_I will solve the problem and produce a nearly constant gain over the range of relevant wind speeds:

$$\bar{K}_p = \frac{2J\ \omega_{wt\_ref}\xi\omega_n}{n_g}\left[-\frac{\partial P}{\partial \beta}\right]^{-1} GK(\beta) \tag{3.8}$$

and

$$\bar{K}_I = \frac{J\ \omega_{wt\_ref}\omega_n^2}{n_g}\left[-\frac{\partial P}{\partial \beta}\right]^{-1} GK(\beta) \tag{3.9}$$

$$GK(\beta) = \frac{1}{1+\frac{\beta}{\beta_K}} \tag{3.10}$$

where β_K is a rotor-collective blade-pitch angle at which the pitch sensitivity has doubled from its value at the rated operating point, and $GK(\beta)$ is the dimensionless gain-correction factor [23] which depends on the blade-pitch angle.

In order to realise the gain-scheduled PI blade-pitch controller, Jonkman et al. (2009) [41] used the blade-pitch angle from the previous controller time step to calculate the gain-correction factor at the next time step.

3.4.1.2 Blade-pitch control system using the dynamic behaviour of the system

Hansen et al. (2005) [23] assumed that the dynamic behaviour of the system could be approximated with second-order system behaviour and, then, the parameter design of the PI speed controller could be based on the transient response analysis for a second-order system:

$$\frac{K_{2nd}}{J_{2nd}s^2 + D_{2nd}s + K_{2nd}} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{3.11}$$

where J_{2nd}, K_{2nd} and D_{2nd} denote the system inertia, stiffness and damping (for assumed second-order system) respectively. Hence, the natural frequency and the damping ratio can be expressed as:

$$\omega_n = \sqrt{\frac{K_{2nd}}{J_{2nd}}} \tag{3.12}$$

$$\xi = \frac{D_{2nd}}{2K_{2nd}}\omega_n \tag{3.13}$$

In the literature, two main parameterisations of the PI controller are applied:

- ***PI controller - parameterisation I***:

$$y = (K_p + \frac{K_i}{s})u \tag{3.14}$$

From equations (3.6), (3.9), (3.12) and (3.13), the proportional and integral gain can be expressed as:

$$K_p = \frac{\omega_{wt\_ref} D_{2nd}}{n_g}\left[-\frac{\partial P}{\partial \beta}\right]^{-1} \tag{3.15}$$

$$K_I = \frac{\omega_{wt\_ref} K_{2nd}}{n_g}\left[-\frac{\partial P}{\partial \beta}\right]^{-1} \tag{3.16}$$

In this PI controller, the integral gain K_I is proportional to the stiffness K_{2nd} of the system, whilst the proportional gain K_P is proportional to the damping of the system D_{2nd}. It should be noted that when this parameterisation is used, both controller parameters contain gain scheduling, i.e. they vary with the sensitivity function $[\frac{\partial p}{\partial \beta}]^{-1}$.

- ***PI controller - parameterisation II:***

$$y = K_p(1 + \frac{1}{sT_i})u \tag{3.17}$$

Then, the integral time is expressed as:

$$T_i = \frac{K_p}{K_I} = \frac{D_{2nd}}{K_{2nd}} \tag{3.18}$$

This parameterisation of the PI controller is used typically in control systems. It should be noted that here only one parameter, i.e. the proportional gain K_P, has a proportional variation with the reciprocal sensitivity function $[\frac{\partial p}{\partial \beta}]^{-1}$, whilst the

integral time T_i could be determined directly based on design parameters (damping D_{2nd} and stiffness K_{2nd} of the system).

As discussed in the previous section, constant values of K_P and K_I will not give the desired result. A gain correction factor, $GK(\beta)$ multiplied with both K_P and K_I will solve the problem and produce a nearly constant gain over the range of relevant wind speeds.

3.4.1.3 Proportional pitch controller

In order to control the wind turbine electric power output, whilst avoiding significant loads and maintaining the rotor speed within acceptable limits, B. Boukhezzar et al. (2007) [38] recommended a proportional pitch controller (see Figure 3.7). The operating point of the blade-pitch angles' perturbation about their operating point is expressed as:

$$\Delta\beta = K_p e_\omega \tag{3.19}$$

where:

$$e_\omega = \omega_{m\_ref} - \omega_m \tag{3.20}$$

The proportional gain is:

$$K_P = \frac{\Delta\beta}{\omega_{m\_ref} - \omega_m} \tag{3.21}$$

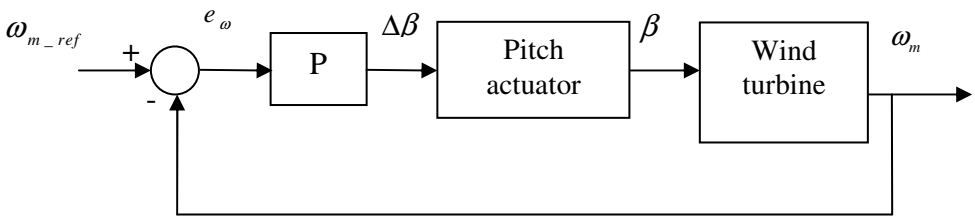


Figure 3.7: Flowchart of the P controller

3.4.2 Individual blade-pitch control

In individual blade-pitch control, the loads of blades are endured by an individual mechanism; consequently, each blade has its own pitch-control mechanism and is driven individually. If one of the mechanisms failed, the others could continue to work. Based on this characteristic, the individual pitch control could be used to reduce the flap of the blade, and, therefore, it has some advantages over the collective one. An individual blade-pitch control could be realised through two modes: the hydraulic mode and the motor mode.

Hongwei et al. (2006) [42] proposed piecewise PID control with weight-number distribution arithmetic (see Figure 3.8). This meant that, after obtaining $\Delta\beta$ based on the collective blade control, weight numbers are allocated by referring to the azimuth angles to reduce the flap of the blades. Generally, the weight numbers are allocated according to the wind speeds on each blade of the wind turbine. The higher the speed is, the bigger the weight number is and the higher the change in the pitch angle.

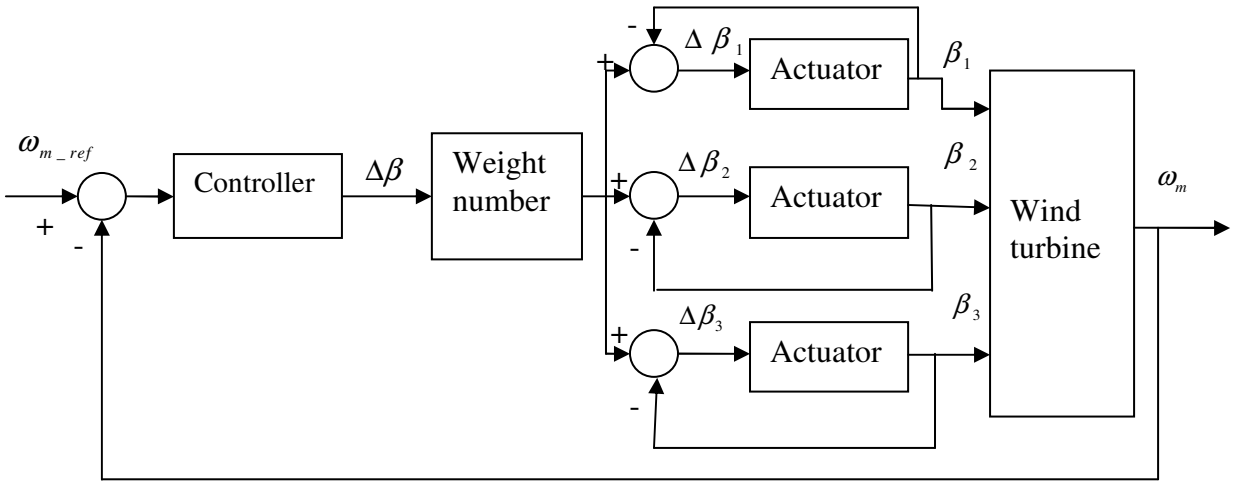


Figure 3.8: Flowchart of the structure of individual control

For large-scale wind turbines it is possible to disregard the random variables of wind speed on the long blades. However, the altitude's effects on the blades are notable. We used equation (3.23), deduced from equation (3.22), to calculate the average wind speeds on the blades as the speeds at the centre of the blade:

$$\frac{V_H}{V_0} = \left(\frac{H}{H_0} \right)^n \tag{3.22}$$

where V_0 is the wind speed on the point of the blade whose distance from the ground is H_0, and V_H is the wind speed on the point whose distance from the ground is *H*. *n* is a parameter ranging from 0.1-0.4 and is decided by the degree of the roughness of the ground.

$$\frac{v_i}{v_0} = \left[1 + \frac{R}{2H_o} \sin\left[\theta + 120^o (i-1) \right] \right]^n \tag{3.23}$$

where *R* represents the length of the blades, θ is the azimuth angle of the first blade, and i=1,2,3 denotes the number of the blades. Therefore, the weight number k_i is calculated by using the following formulation:

$$k_i = \frac{3\left[\frac{v_i}{v_0} \right]}{\sum_{i=1}^{3} \left[\frac{v_i}{v_0} \right]} \tag{3.24}$$

After the weight number allocation, the change, which should have been transferred to the actuator mechanism, is:

$$\Delta\beta_i = k_i \Delta\beta \tag{3.25}$$

3.5 Determination of the PI pitch controller gains

Variable-speed variable-pitch wind turbine modes take into account rotational speed and power limitations. These modes are:

1) Operating at minimal value (near to cut-off in wind speed), the tip speed ratio could be optimised by pitch control.

2) Variable-speed fixed-pitch operation: the operating point describes the optimal regime's characteristic. Here, a torque control scheme to optimise the power is used.
3) Fixed-speed variable-pitch operation: the rotational speed limitation at its rated value. In this mode, the pitch angle controller to achieve power limitation is used.

3.5.1 Selection of the operating point

As shown in Table 3.1, the nominal parameters of a 5MW wind turbine were used [16], with an assumed wind speed of 15 m/s. From Figure 3.2, this wind speed is above the rated speed, and, therefore, the operating point is in mode 3. Then, rotational speed must be equal to the rated rotor speed (ω_{wt_ref}). Consequently, the desired constant speed of the turbine is 1.87 rad /s. By using Figure 3.9 and Figure 3.10, λ and C_p operating points were selected. As selected from the power coefficient curve of Figure 3.1, the pitch angle is 9.65^o. By repeating the same steps for each wind speed, it was possible, as shown in Table 3.2, to calculate the optimal values of λ, C_p and pitch angle for mode 3.

Table 3.1: Parameters of the 5MW wind turbine and controller

Description	Parameter	Value
Rated turbine power	P_n	*5MW*
Turbine blade length	R	*55m*
Gearbox ratio	n_g	*60.88*
Air density	ρ	*1.225 kg/m^3*
Reference generator speed	ω_{m_ref}	*113.85(rad/s)*
Cut-in wind speed	v_{cin}	*3 m/s*
Cut-out wind speed	v_{cout}	*25 m/s*

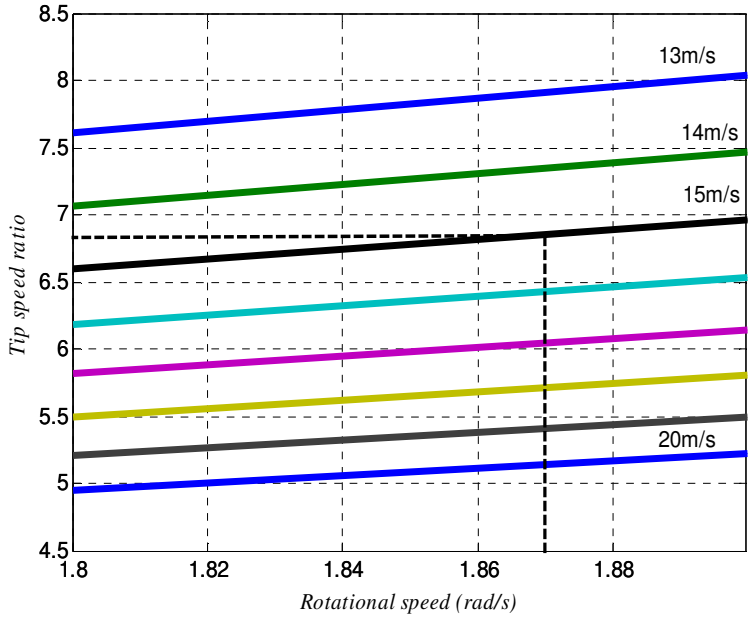


Figure 3.9: Tip speed ratio *vs.* wind turbine rotational speed shows operating mode 3, fixed-rotational speed and variable-pitch operation

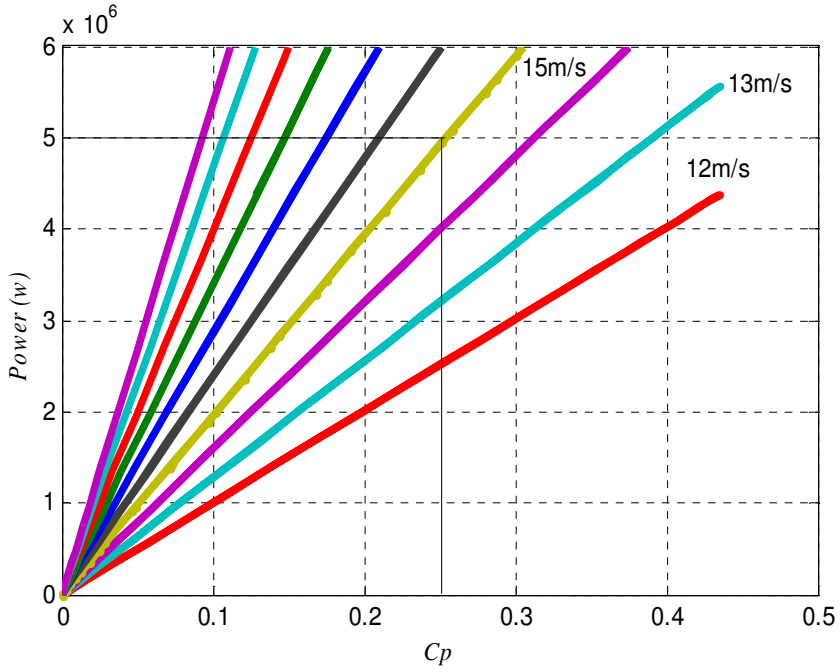


Figure 3.10: Wind turbine output power *vs.* C_p shows operating mode 3, fixed-rotational speed and variable-pitch operation

3.5.2 Theoretical method to calculate PI controller gains

Figure 3.11 illustrates the PI controller's output signal is β_d, which contains, also, the actuator's transfer function obtained from equation (3.4). Then, the PI controller and desired pitch angle can be expressed as follows:

$$\beta_d = K_p e_\omega + K_i \int e \, dt \tag{3.26}$$

where

$$e_\omega = \omega_{m\_ref} - \omega_m \tag{3.27}$$

To find the solution, let:

$$x = K_i \int e_\omega dt \tag{3.28}$$

or

$$\frac{dx}{dt} = K_i e_\omega \tag{3.29}$$

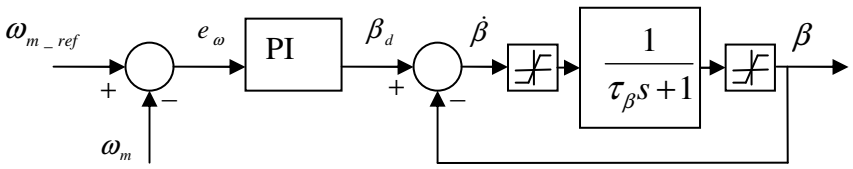


Figure 3.11: PI Pitch controller

From equations (3.26) and (3.28), the partial derivative of β_d, with respect to e_ω, is expressed as follows:

$$\frac{d\beta_d}{de_\omega} = Kp + \frac{dx}{de_\omega} = Kp + \frac{dx/dt}{de_\omega/dt} = Kp + K_i \frac{e_\omega}{\frac{de_\omega}{dt}} \tag{3.30}$$

For an adjustable-slip asynchronous generator, the variation range of e_ω is very small. Moreover, K_P is far greater than K_i. Hence, equation (3.30) can be simplified as follows:

$$K_p = \frac{d\beta_d}{de_\omega} \tag{3.31}$$

Table 3.2: Wind turbine characteristics

Power optimisation					
v *(m/s)*	λ	C_p	β	K_{opt}	ω_{wt_ref} *(rad/s)*
3	11	0.43	2	1.3842	0.5984
4	9.9	0.44	2	1.9242	0.7293
5	10	0.43	2	1.796	0.9163
6	9.9	0.44	2	1.9242	1.0846
7	10	0.43	2	1.6726	1.3277
8	9.9	0.44	2	1.9242	1.4399
9	11.43	0.44	2	1.2513	1.87
10	10.29	0.44	2	1.7161	1.87
11	9.4	0.43	2	2.2568	1.87
12	8.6	0.41	2	2.8219	1.87
Power limitation					
v *(m/s)*	λ	*cp*	β	K_p	ω_{wt_ref} *(rad/s)*
13	7.9	0.39	2	24.09	1.87
14	7.3	0.31	5.85	16.26	1.87
15	6.8	0.25	9.65	20.51	1.87
16	6.4	0.21	13	52.2	1.87
17	6	0.17	15.75	33.48	1.87
18	5.7	0.15	18.16	131.25	1.87
19	5.4	0.13	20.23	146.57	1.87
20	5.1	0.11	22.22	47.23	1.87
21	4.8	0.09	24.03	21.8	1.87
22	4.6	0.08	25.52	27.95	1.87
23	4.4	0.07	26.9	29.46	1.87
24	4.2	0.06	28.21	24.86	1.87
25	4.1	0.06	29.15	150.38	1.87

where $d\beta_d = \beta_d$ with initial value β_{d_in}=0. To find the direct relation between β and β_d, the inner closed loop was reduced for the actuator, as shown in Figure 3.13, to the forward path and assumed τ_β=1s. Consequently, the following transfer function is obtained:

$$\frac{\beta}{\beta_d} = \frac{1}{s+2} \tag{3.32}$$

In the steady state, s $\Rightarrow$ 0, and $\beta_d = 2\beta$. Then substitute $d\beta_d = \beta_d$ in equation (3.31), K_p is:

$$K_p = \frac{2\beta}{\omega_{m\_ref} - \omega_m} \tag{3.33}$$

And substitute equation (3.33) in equation (3.30), then K_i is:

$$K_i = \frac{1}{e_\omega} * \left(\frac{d\beta_d}{de_\omega} - \frac{2\beta}{\omega_{m\_ref} - \omega_m}\right) * \frac{\partial\omega}{\partial t} \tag{3.34}$$

From equations (3.33) and (3.34), the integral coefficient (K_i) equals zero since the middle part of equation (3.34) equals zero in the steady state. Then the controller is summarised as:

$$K_p^j = \frac{2\beta^j}{e_\omega^j}, \; K_i^j = 0$$

where β^j and e_ω^j are blade-pitch angle and error at wind speed v^j. By using the plot of pitch angle β *vs.* wind speed (Figure 3.12), it is possible to design scheduling as illustrated in Figure 3.13.

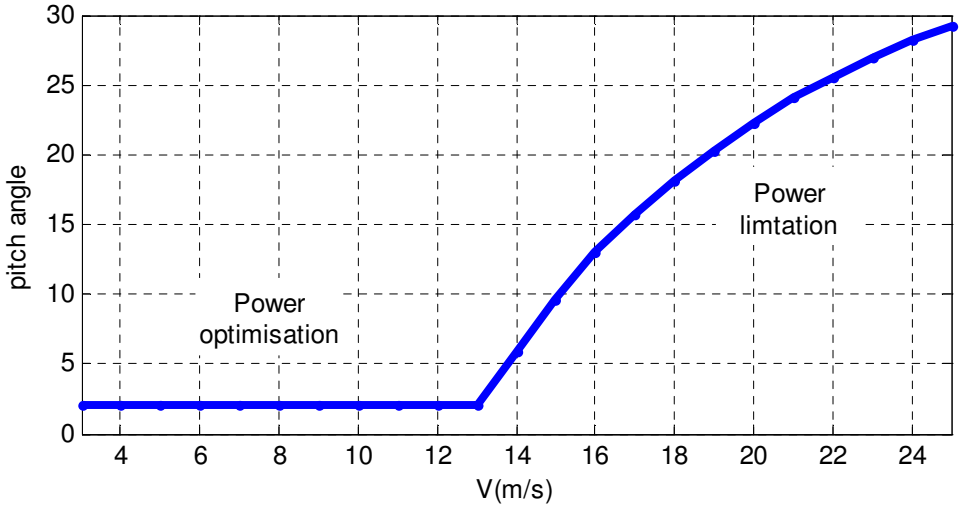


Figure 3.12: Pitch angle β *vs.* wind speed. The value of β is between 2°-30°

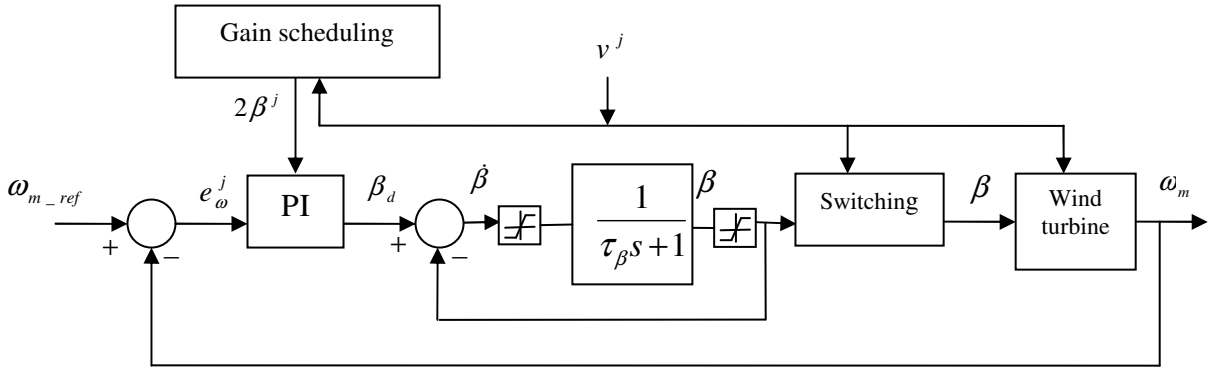


Figure 3.13: Block diagram of the pitch angle PI controller based on the theoretical method. Simulink models for switching (change from electrical torque controller to blade-pitch controller) and WT are presented in Figures B.12 and B.13, respectively

3.5.3 Simulation-based method to calculate PI gains

Above the rated wind speed, the rotor power must be kept at the design limits by using the PI pitch angle controller. Here, it is necessary to compute the PI gains of the pitch angle controller for all the system's operating points.

From Figure 3.11, the relationship between error e_ω and pitch angle β is:

$$\frac{\beta(s)}{E_\omega(s)} = (K_p + \frac{K_i}{s})\frac{1}{s+2} \tag{3.35}$$

Since values of β and e_ω were selected from Table 3.2 and the wind turbine model respectively, it was possible to calculate K_P and K_i by using simulations to shape the responses in equation (3.35).

Figure 3.14 shows the block diagram, which was implemented to calculate K_P and K_i by inserting data of wind turbine as input and gains as output. We implemented the simulation by using Matlab/Simulink software. The generator speed ($\omega_m = n_g \omega_{wt}$) is the feedback and the rated generator speed, ω_{m_ref} is a set value. The error is input to the controller, which commands a change in the blade-pitch angle. The new requested pitch angle is $\beta_{new} = \beta_d - \beta$. The actuator operates on a pitch rate command. The pitch rate is determined from the difference between the desired pitch angle and the measured blade-pitch angle. By using this method, the PI gains were obtained, as shown in Figure 3.14, and achieved desired rotor speeds and output power. Table 3.2 was used to find control gains at different wind speeds (13, 14 …25 m/s). The value of K_i was very small (K_i <<1/10000).

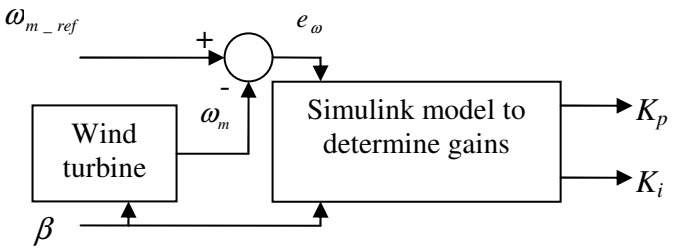


Figure 3.14: Block diagram to calculate K_p and K_i. Simulink model to determine gains is presented in Figure B.23

For example, Figure 3.15 shows the values of K_p and K_i at wind speed 15 m/s. Figure 3.16 shows the response of the output pitch angle at different operating points (wind speed = 12, 15, 18, 21 m/s). Here, it should be noted that, when the wind speed 12 m/s is below the rated value, the controller should switch to electrical torque controller.

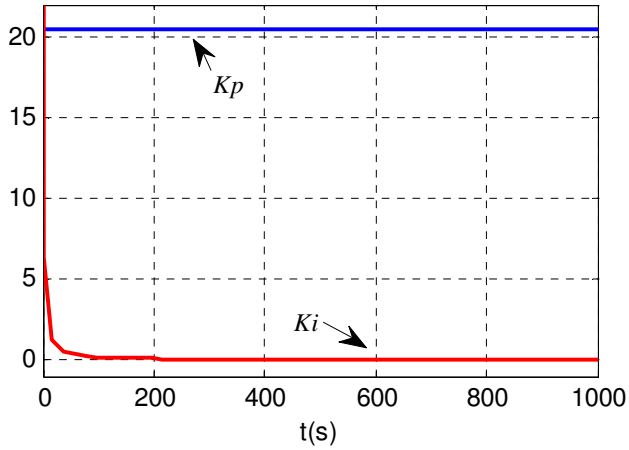


Figure 3.15: Values of K_p and K_i at wind speed 15 m/s

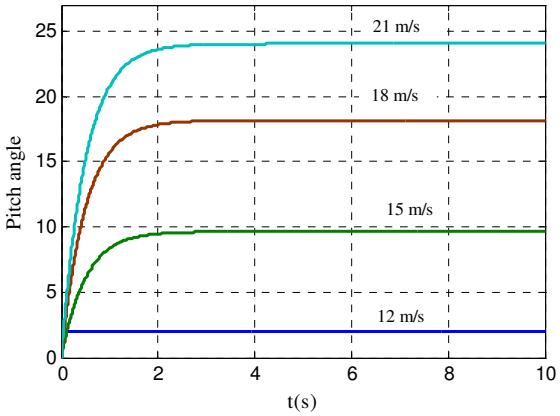


Figure 3.16: Shows the simulation output of pitch angle at different operating points (wind speed v = 12, 15, 18, 21 m/s)

3.6 Generator-torque controller

For the 5MW wind turbine, a conventional variable-speed, variable blade-pitch were chosen. In such wind turbines, the conventional approach for controlling power-production operation relies on the design of two basic control systems: a generator-torque controller and a full-span rotor-collective blade-pitch controller. The two control systems are designed to work independently, for the most part, in the below-rated and above-rated wind-speed range, respectively. In this section a generator-torque controller will be designed to maximise power capture below the rated wind speed.

3.6.1 Model strategy of the generator

In section 2.6.1 the stator voltage vector is oriented to the q-axis of the reference frame. Therefore, it can be seen that the stator voltage vector and the order of the DFIG model decreases from fourth to second order which is beneficial in simplifying the DFIG's excitation control system (see Figure 3. 17).

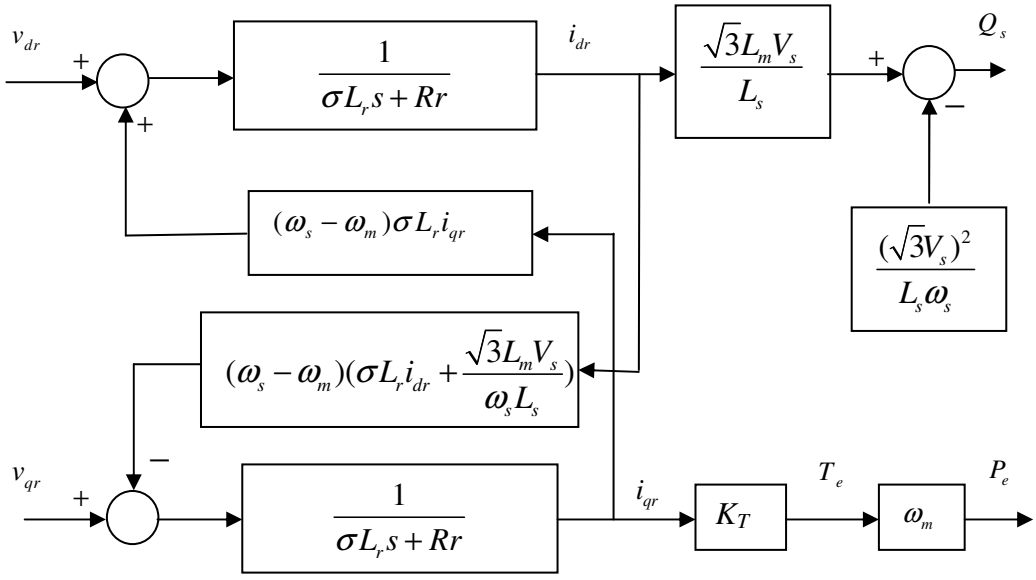


Figure 3. 17: Block diagram of the DFIG model

3.6.2 Control strategy of DFIG

In order to realise the maximum power point tracking, the electric torque was considered to be a reference. Under the current-mode control, the electric torque is proportional to the q-axis rotor current i_{qr}. Consequently, the reference value i_{qr} for the current inner loop could be derived from T_e directly as is discussed in section 2.6.1 of the previous chapter as:

$$i_{qr\_ref} = -\frac{L_s \omega_s}{n_p L_m V_s \sqrt{3}} T_{e\_ref} = -K_T T_{e\_ref} \tag{3.36}$$

Figure 3. 17 depicts the cascaded control scheme under torque-mode control. A PI controller is the controller in the torque outer loop. Under current-mode control, a PI controller is the controller in the current inner loops.

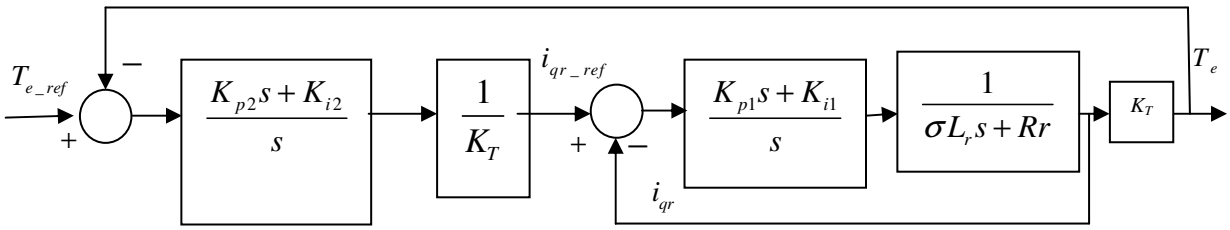


Figure 3.18: Control structure under current and electric torque

3.6.3 Reference electrical torque

When the wind speed is below rated, optimising the wind turbine's power output is achieved by using the torque control scheme for a variable-speed wind [17]. In this case, the following equations hold:

$$T_{\text{e\_ref}} = \frac{P_{wt(\max)}}{\omega_{\text{m}}} \tag{3.37}$$

or

$$T_{e\_ref} = \frac{0.5\rho\pi R^2 C_{p\_\max} (\frac{R\omega_{wt}}{\lambda_{opt}})^3}{\omega_m} \tag{3.38}$$

where T_{e_ref} is the reference electrical torque, $C_{p\_\max}$ is the maximum power coefficient and λ_{opt} is the tip speed ratio at $C_{p\_\max}$.

Since in the steady state, $\omega_m = n_g \omega_{wt}$, the optimal gain is defined as:

$$K_{opt} = \frac{0.5\rho\pi R^5 C_{p\_\max}}{\lambda_{opt}^3 n_g^3} \tag{3.39}$$

Therefore, equation (3.38) can be rewritten as:

$$T_{e\_ref} = K_{opt}\omega_m^2 \tag{3.40}$$

It should be noted that the transmission friction losses were not considered. From equation (3.39), it is clear that the optimal gain varies from one turbine to another, even when they both have the same rated power. Furthermore, these gains could be changed during a turbine's lifetime. For example, dirt on blades or a slightly damaged blade can reduce the aerodynamic efficiency.

3.6.4 IMC-based PI controller

Internal Model Control (IMC) is a commonly used technique that provides a transparent mode for the design and tuning of various types of control. In this section, by using the IMC-based PI controller [43, 44], it was possible to design the controller in current and torque loops in order to achieve good static and dynamic performance. The IMC based PID structure uses the process model as in IMC design. In the IMC procedure, the controller *Q(s)* is directly based on the invertible part of the process transfer function. The IMC results in only one tuning parameter, which is filter tuning parameter.

Here, as shown in Figure 3.18, the IMC-based PI design procedure was applied for the control structure under current and torque.

I- Given current model:

$$G_{iqr}(s) = \frac{1}{\sigma L_r s + R_r} \tag{3.41}$$

Transfer function for PI controller:

$$C_{idr}(s) = C_{iqr}(s) = \frac{K_p s + K_i}{s} \tag{3.42}$$

By applying the IMC, the equivalent standard feedback controller is obtained using the transformation:

$$C_{iqr}(s) = \frac{Q_{iqr}}{1 - Q_{iqr} G_{iqr}} = \frac{\sigma L_r s + R_r}{\lambda_f s} \tag{3.43}$$

where

$$Q_{iqr} = inv(G_{iqr}).\frac{1}{\lambda_f s + 1} \tag{3.44}$$

λ_f is the filter-tuning parameter to vary the speed of the response of a closed-loop. Then, the result is compared with the PI controller transfer function and, hence, the PI tuning parameters are:

$$K_p = \frac{\sigma L_r}{\lambda_f} \text{ and } K_i = \frac{R_r}{\lambda_f}$$

To confirm that the current inner loop's response is faster than that of the torque outer loop, λ_f =0.1 was selected; then $K_p = 0.97$ and $K_i = 0.018$ (see Figure 3.19).

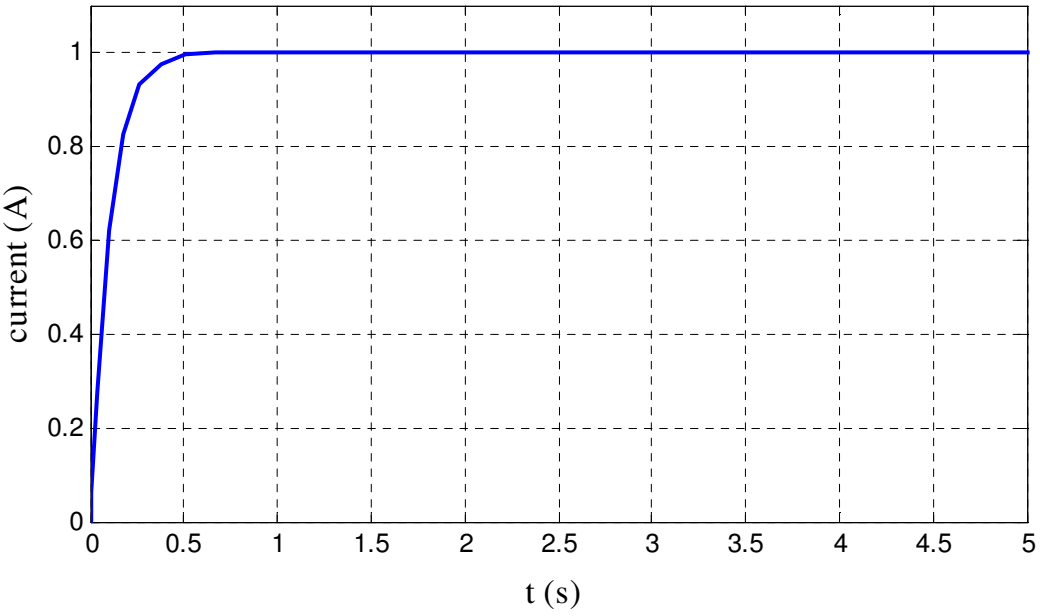


Figure 3.19: Simulation of IMC-based PI current model when λ_f =0.1

II- Given torque model:

From figure 3.18 the transfer function for torque is:

$$G_{T_e}(s) = \frac{1}{K_T} . \frac{\dfrac{\sigma L_r s + R_r}{0.1s(\sigma L_r s + R_r)}}{1 + \dfrac{\sigma L_r s + R_r}{0.1s(\sigma L_r s + R_r)}} . K_T = \frac{1}{0.1s + 1} \qquad (3.45)$$

By applying the IMC, the equivalent standard feedback controller is obtained using the transformation:

$$C_{T_e}(s) = \frac{Q_{T_e}}{1 - Q_{T_e} G_{T_e}} = \frac{0.1s + 1}{\lambda_f s} \tag{3.46}$$

where

$$Q_{T_e} = inv(G_{T_e}).\frac{1}{\lambda_f s + 1} \tag{3.47}$$

Therefore, the PI-tuning parameters of the torque closed loop are:

$$K_p = \frac{0.1}{\lambda_f}$$

$$K_i = \frac{1}{\lambda_f}$$

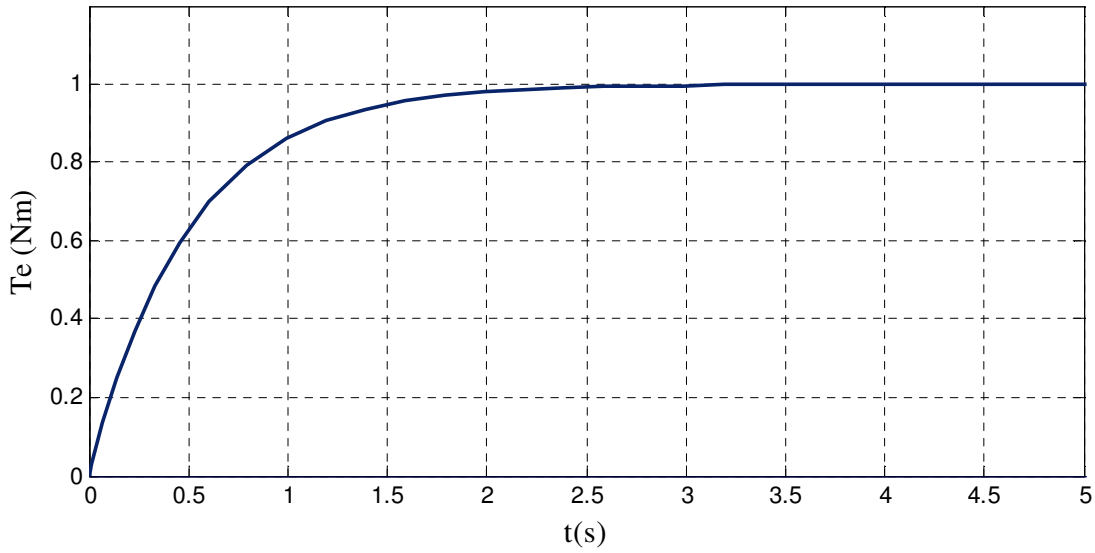


Figure 3.20: Simulation of IMC-based PI torque model when $\lambda_f = 0.5$

From Figure 3.20 it can be seen that when $\lambda_f = 0.5$, the gains of PI torque controller are $K_p = 0.2$ and $K_i = 2$. The simulation results in Figure 3.21 demonstrate that the torque controller is accomplished.

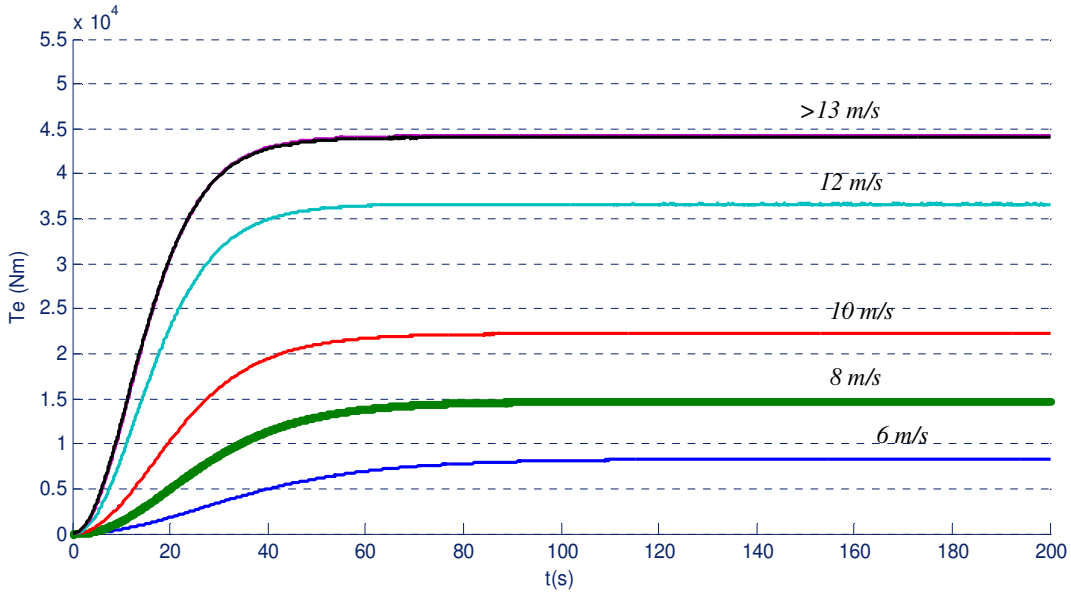


Figure 3.21: Electrical torque response at different wind speeds (6, 8, 10, 12 m/s and above-rated speed)

3.7 Simulation results for a non-linear wind turbine model

In order to control the pitch angle and output power for non-linear wind turbine models, gain-scheduling was implemented. This uses its input value to select a case condition, which determines the subsystem that should be executed to give the K_P value as output as shown in Figure 3.22 and B.10. Figure B.12 illustrates that the switch block receives wind speed as a single input to change to the blade pitch control system and/or a change in the electrical generator torque control loop.

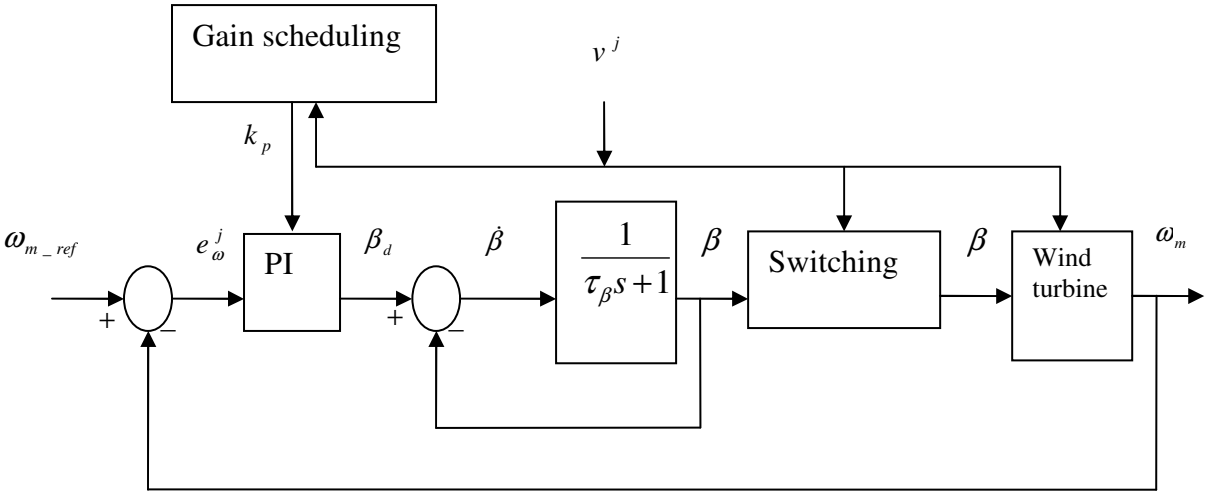


Figure 3.22: Block diagram of the pitch angle PI controller based on the simulation method. Simulink models for gain scheduling, PI controller, switching and wind turbine are presented in Figures B.11, B.12 and B.13, respectively

The simulation output results for turbine characteristics and tracking characteristics, presented in Figure 3.23, and the operating regimes for C_p-λ shown in Figure 3.24, demonstrate that power optimisation and power limitation are realised. Figure 3.25 shows output power from the wind rotor *vs.* wind speed.

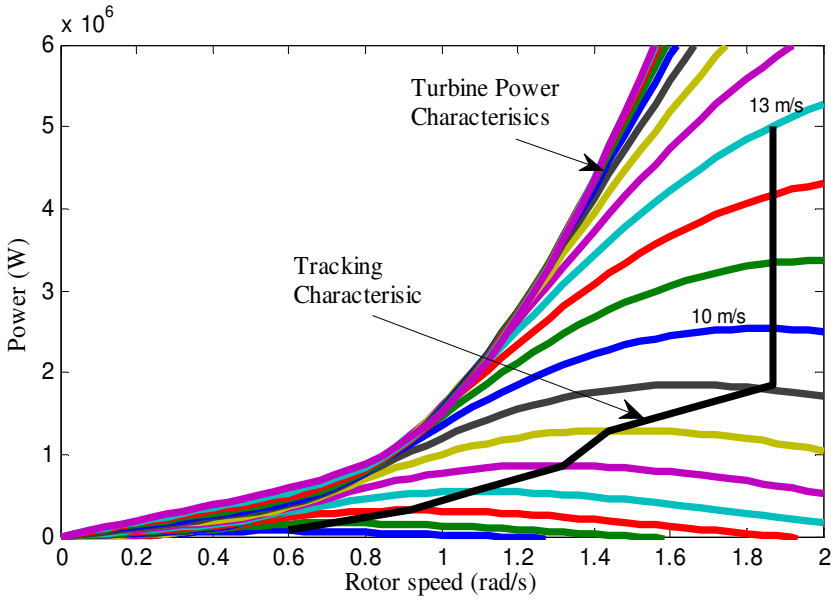


Figure 3.23: Illustrative turbine characteristics and tracking characteristics achieved by using the electrical torque controller and blade-pitch angle controller

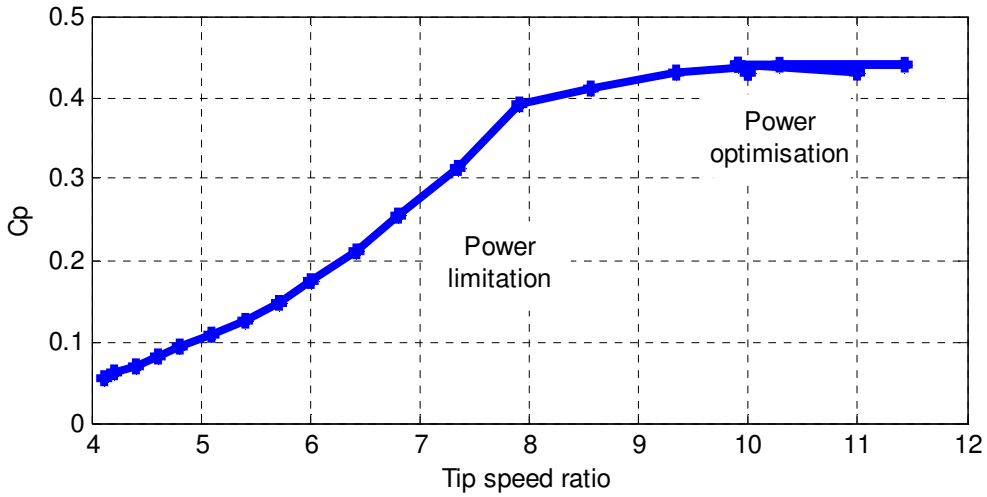


Figure 3.24 Illustrative operating regimes for Cp-λ

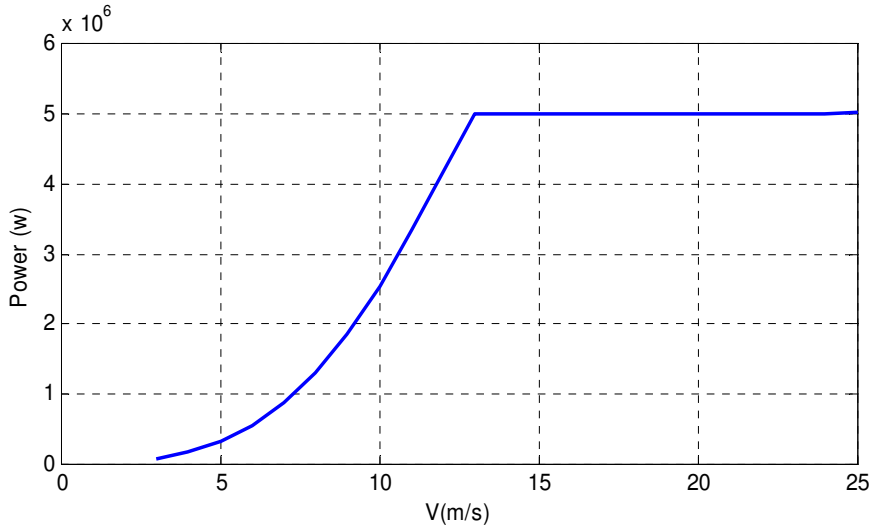


Figure 3.25: Output power from wind rotor *vs.* wind speed

3.8 Conclusions

This chapter focused on the design of a PI controller for a 5MW wind turbine. Firstly, two PI-based control design schemes were proposed for the blade-pitch angle controller in a variable-speed wind turbine; one analytical method and one simulation-based method to calculate the PI gains. The simulation results demonstrated good performance for both proposed PI schemes.

Secondly, the IMC-based PI design procedure was applied for the control of current and electric torque. Figures B.15 and B.14 for subsystems to calculate the power coefficient and optimal tip speed ratio enabled the design of a rotor torque model without feedback of the rotational speed of the rotor, such that this non-linear model had similar characteristics to that of a real wind turbine (the wind turbine subsystem converts wind speed into aerodynamic torque) (see Figure B.13 and B14).

4 Model-based Fault Detection and Isolation Methods for Wind Turbines

4.1 Introduction

Faulty components in wind turbine can cause high loses in energy production and possible damage to the wind turbines. The losses may be higher for offshore wind farms. This decreases the reliability and increases the cost of maintenance of the wind turbines. Therefore, the aim of the wind turbine fault monitoring is to avoid abnormal event progression and reduces productivity loss, system breakdowns and damage. It increases safety and reliability of the system to achieve higher performance.

Model-based Fault Detection and Isolation Methods surveyed in literature can be classified into two general categories, quantitative and qualitative methods. In quantitative method, the understanding is expressed in terms of mathematical functional relationships between the inputs and outputs of the system in the form of system descriptions. In qualitative method, the relationships are expressed in terms of qualitative functions between different components of the system. This approach usually depends upon the knowledge from experts in both the normal and fault cases.

In this chapter, a quantitative model based method is proposed for early fault detection and diagnosis of wind turbines. The method is based on designing an observer using a model of the system. For application to the wind turbines, a simplified state space version of the wind turbine model is derived for design an observer. The observer innovation signal is monitored to detect faults. The fault detection system is designed and optimised to be most sensitive to system faults and least sensitive to system disturbances and noises. A multi-objective optimization

method is then employed to solve this dual problem. Simulation results are presented to demonstrate the performance of the proposed method.

4.2 Observer-based fault detection

The basic design of observer-based fault detection is to compare the actual measurements (y) with the output provided by an observer ($\hat{y}$), so that the residual depends on the difference between y and $\hat{y}$ as shown in Figure 4.1.

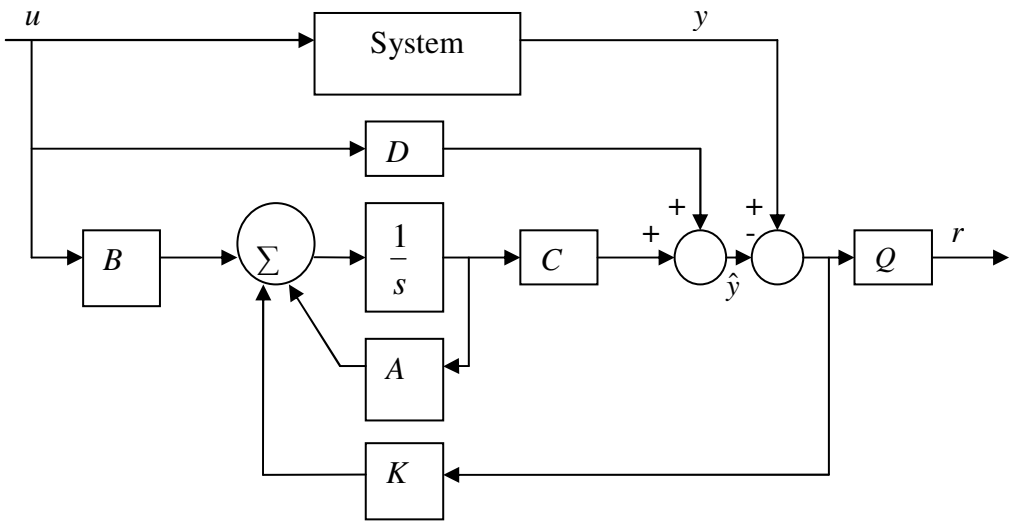


Figure 4.1: Block diagram of observer-based residual generation

The system described by equation (4.1) is used to design an observer. The mathematical description of the observer is the same as that of the system except that the observer has an additional term, the gain K, which continuously corrects the system output and improves the state estimates. The observer is defined in equation (4.2).

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{4.1}$$

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + K(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t) + Du(t) \end{aligned} \tag{4.2}$$

where x, u, y are the state, input and output of the system of dimensions n, m and p, respectively and A, B, C and D are system matrices of appropriate dimensions.

To define the error, equation (4.2) is subtracted from equation (4.1):

$$\begin{aligned} y(t)-\hat{y}(t) &= C(x(t)-\hat{x}(t)) \\ \dot{x}(t)-\dot{\hat{x}}(t) &= A(x(t)-\hat{x}(t))-K(y(t)-\hat{y}(t)) = (A-KC)(x(t)-\hat{x}(t)) \end{aligned} \tag{4.3}$$

Defining the difference between $x(t)$ and $\hat{x}(t)$ as the state error vector (residual), *e(t)*, the dynamic error can be written as:

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) = (A-KC)e(t) \tag{4.4}$$

For fault monitoring purposes, a weighted residual is defined as follows:

$$r(t) = QCe(t) \tag{4.5}$$

where Q is the residual weighting factor.

Equation (4.10) illustrates the dynamic behaviour of the innovation signal and this is governed by the eigenvalues of the matrix *(A-KC)*.

If the matrix *A-KC* is stable, the error will tend to zero or a constant. If the eigenvalues are chosen in such a way that the dynamic behaviour of the error is asymptotically stable and adequately fast, then any error will tend to zero with sufficient speed. This is possible by choosing an appropriate value for K to achieve the stability when the system is completely observable. The observability can be examined using equation (4.6).

$$\mathrm{O} = \begin{vmatrix} C \\ CA \\ CA^2 \\ . \\ . \\ CA^{n-1} \end{vmatrix} \neq 0 \tag{4.6}$$

4.2.1 Observer gain matrix

It is assumed the matrices *A, B, C* and *D* are identical for both the observer and the system, so if there is any difference, the dynamics of the observer error are no longer

governed by equation (4.4). It is necessary to find K, so that the observer is stable and the error remains acceptably small.

4.2.1.1 Transformation approach to find state observer gain matrix

By using this method, the observer gain matrix K can be calculated by the following equation [85]:

$$K = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = \left(\begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \right)^{-1} \begin{bmatrix} \alpha_n - a_n \\ \alpha_{n-1} - a_{n-1} \\ \vdots \\ \alpha_1 - a_1 \end{bmatrix} \tag{4.7}$$

where the characteristic polynomial for matrix A is given by:

$$|sI - A| = s^n + a_1 s^{n-1} + a_2 s^{n-2} \cdots + a_{n-1} s + a_n \tag{4.8}$$

and the desired eigenvalues for closed loop are:

$$(s - p_1)(s - p_2) \cdots (s - p_n) = s^n + \alpha_1 s^{n-1} + \alpha_2 s^{n-2} + \cdots + \alpha_{n-1} s + \alpha_n \tag{4.9}$$

When the matrix of the system is in the observable canonical form, then the gain matrix can be obtained by the following equation:

$$K = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} \alpha_n - a_n \\ \alpha_{n-1} - a_{n-1} \\ \vdots \\ \alpha_1 - a_1 \end{bmatrix} \tag{4.10}$$

4.2.1.2 Ackermann's formula for determination of the observer gain matrix

Ackermann's formula to find the matrix K is given by equation (4.11).

$$K = \left[A^n + \alpha_1 A^{n-1} + \alpha_2 A^{n-2} + \cdots + \alpha_{n-1} A + \alpha_n I \right] \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{4.11}$$

The gain matrix *K* depends on the desired characteristic equation; to make sure the observer error is very small; the observer poles must be five times faster than the controller poles. In the case where the sensor noise is considerable, it is possible to select the observer poles to be two times slower than the controller poles, so that the bandwidth of the system will become lower and smooth the noise. In this case, the system response will be strongly influenced by the observer poles. If the observer poles are located to the right of the controller poles in the left-half S plan, then the system response will be dominated by the observer poles [85].

In the design of the observer, it is important to obtain a number of observer gain matrices based on different desired characteristic equations. Then, it is possible to select the best from the viewpoint of overall system performance by using simulation. The selection of the best observer gain matrix *K* depends on a compromise between response and sensitivity to disturbances and noises.

4.2.2 Residual generation

Assume the system is fully observable. The system dynamics with faults and disturbance models can be written as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + R_1 f(t) + d(t) \\ y(t) &= Cx(t) + Du(t) + R_2 f(t) \end{aligned} \tag{4.12}$$

where *f(t)* represents the fault vector and is considered to be an unknown function of time. The vector *d(t)* is the disturbance vector and can be written as in equation (4.15). The matrices R_1 and R_2 are fault distribution matrices; they can be determined

if one has defined which faults are to be diagnosed. For example, sensor and actuator fault matrices can be represented as:

$$R_1 = \begin{cases} 0 \\ B \end{cases} \begin{array}{l} sensor\ faults \\ actuator\ faults \end{array} \tag{4.13}$$

$$R_2 = \begin{cases} I_m \\ D \end{cases} \begin{array}{l} sensor\ faults \\ actuator\ faults \end{array} \tag{4.14}$$

$$d(t) = \Delta A x(t) + \Delta B u(t) \tag{4.15}$$

The residual generator studied in this section, as explained in Figure 4.2, is based on an observer. The principle is to estimate the system output from the measurements using an observer. The weighted estimation error is used as a residual. The flexibility in this selection, the observer gain and the weighting matrix give a choice to achieve good detection performance.

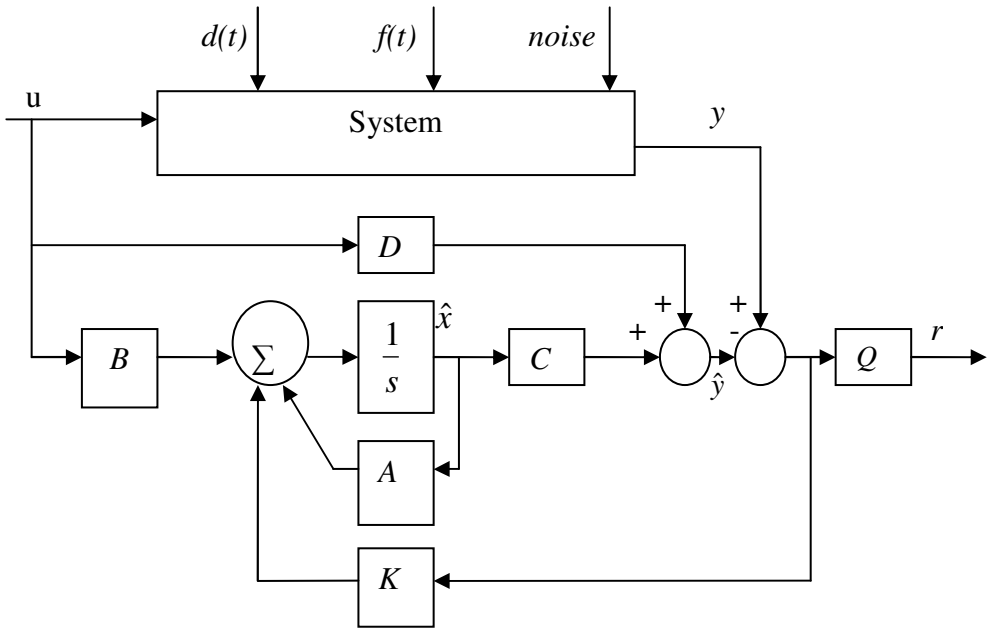


Figure 4.2: Residual generation via a full order observer

When this residual generator is applied to monitoring a system represented by equation (4.12) and to define error, equation (4.2) is subtracted from equation (4.12)

then the estimation error and residual are obtained, as in equations (4.16) and (4.17) respectively, and

$$\dot{e} = \dot{x}(t) - \dot{\hat{x}}(t) = (A - KC)e(t) + d(t) + R_1 f(t) - KR_2 f(t) \tag{4.16}$$

$$r(t) = Q[Ce(t) + R_2 f(t)] \tag{4.17}$$

By taking the Laplace transform of equation (4.17), the residual can be written as:

$$\begin{aligned} r(s) &= Q[R_2 + C(sI - A + KC)^{-1}(R_1 - KR_2)]\mathrm{f(s)} + QC(sI - A + KC)^{-1} \\ (d(s) + e(0)) &= G_{rf}(s, K, Q) f(s) + G_{rd}(s, K, Q)(\mathrm{d(s)} + \mathrm{e(0)}) \end{aligned} \tag{4.18}$$

where *e(0)* is the initial value of the state estimation error.

4.2.3 Performance indices in residual generation

Faults and disturbances affect the residual. Therefore, to design good fault detection, it is necessary to maximise the effect of the faults on the residual and minimise the effect of the disturbances on the residual. This means that the effect of the faults must be maximised by maximising the following performance index in the frequency domain [91]:

$$\bar{J}_1(K, Q) = \inf_{\omega \in [\omega_1, \omega_2]} \underline{\sigma}\{[QR_2 + QC(j\omega I - A + KC)^{-1}(R_1 - KR_2)]\} \tag{4.19}$$

and this is equivalent to the minimisation of the following performance index:

$$J_1(K, Q) = \sup_{\omega \in [\omega_1, \omega_2]} \bar{\sigma}\{-[QR_2 + QC(j\omega I - A + KC)^{-1}(R_1 - KR_2)]\ \} \tag{4.20}$$

where $\underline{\sigma}\{.\}$ and $\bar{\sigma}\{.\}$ denote to minimal and maximal singular values.

Similarly, the effect of the disturbances and initial condition on the residual must be minimised by minimising the following performance index:

$$J_2(K, Q) = \sup_{\omega \in [\omega_1, \omega_2]} \bar{\sigma}\{QC(j\omega I - A + KC)^{-1}\} \tag{4.21}$$

Here, only faults and disturbances are considered since noise in the system can affect the residual. To study the effect of noise, it is assumed that $\xi(t)$ and $\eta(t)$ are input and sensor noise signals, respectively. Thus, the system equation will be:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + R_1 f(t) + d(t) + \xi(t) \\ y(t) &= Cx(t) + Du(t) + R_2 f(t) + \eta(t) \end{aligned} \tag{4.22}$$

It can be seen that the sensor noise as well as faults acting through $R_2 f(t)$ affect the system at the same excitation point and, hence, affect the residual in the same way. Therefore, in order to reduce the effect of noise on the residual the following norm should be minimised:

$$\| Q - QC(j\omega I - A + KC)^{-1} K \| \tag{4.23}$$

The minimisation of the above norm contradicts the requirement for maximising the effects of the faults on the residual. Moreover, the frequency ranges of the faults and noise are usually different. For an incipient fault signal, the fault information is contained within a low frequency band since the fault development is slow; however, noise comprises mainly high frequency signals. Based on these observations, the effects of noise and faults can be separated by using different frequency-dependent weighting penalties ($W(jw)$). In this case, the index J_1 is given by:

$$J_1(K,Q) = \sup_{\omega \in [\omega_1, \omega_2]} \bar{\sigma}\{W_1(jw)(-[QR_2 + QC(j\omega I - A + KC)^{-1}(R_1 - KR_2)])\} \tag{4.24}$$

and the minimising effect of noise on the residual, index J_3 is:

$$J_3(K,Q) = \sup_{\omega[\omega 1, \omega 2]} \bar{\sigma}\{W_3(j\omega)Q[I - C(j\omega I - A + KC)^{-1} K]\} \tag{4.25}$$

To maximise the effects of faults at low frequencies and to minimise the noise effect at high frequencies, the frequency-dependent weighting factor $W_1(j\omega)$ should have large magnitude in the low frequency range and small magnitude at high frequencies.

The frequency effect of $W_3(j\omega)$ should be opposite to $W_1(j\omega)$ and can be chosen as $W_3(j\omega)=W_1^{-1}(j\omega)$.

The disturbance and input noise affect the residual in the same way. As both effects should be minimised, the performance index J_2 does not necessarily need to be weighted. However, modelling uncertainty and input noise effects may be more complex for one or more frequency bands. The performance index should reflect this fact and, hence, a frequency-dependent weighting factor must also be placed on $J_2(K,Q)$ in some situations:

$$J_2(K,Q) = \sup_{\omega\in[\omega_1,\omega_2]} \bar{\sigma}\{W_2(j\omega)QC(j\omega I - A + KC)^{-1}\} \tag{4.26}$$

Now, considering the steady state value of the residual:

$$r(\infty) = Q[R_2 - C(A - KC)^{-1}(R_1 - KR_2 -)]f(\infty) - (A - KC)^{-1}d(\infty) \tag{4.27}$$

After the transient period, the residual steady state value plays an important role in fault detection; ideally, it should reconstruct the fault signal. The disturbance effects on the residual can be minimised by minimising the following performance index:

$$J_4(K) = \| (A - KC)^{-1} \| \tag{4.28}$$

When J_4 is minimised, the matrix K is very large and the norm $\|(A-KC)\|$ approaches a constant value. This means that the fault effect on the residual has not been changed by reducing the disturbance effect. This is what is required for good FDI performance.

Four performance indices, $J_1(K,Q)$, $J_2(K,Q)$, $J_3(K,Q)$ and $J_4(K,Q)$, have been defined. To achieve robust fault detection, a multi-objective optimisation problem must be solved. The parameter set to be designed is the observer gain matrix K, which must guarantee the stability of the observer. This leads to a constrained optimisation problem that is difficult to solve. The observer design is a dual problem of the controller design and all techniques in control design can be applied.

4.3 Observer-based fault detection and isolation for a wind turbine

4.3.1 Linear model of the wind turbine

The non-linear model for 5MW wind turbine was designed and demonstrated in Simulink/Matlab and is presented in Chapter 2and Chapter 3. The next step in this study is to design the fault detection and isolation system. Therefore, it is necessary to linearise the non-linear model and determine the state-space model. The linearisation method is performed using the following steps:

1) Determine the operating point equation by solving the non-linear equation in a stationary point.
2) Approximate the non-linear expiration with a first-order Taylor approximation.
3) Subtract the operating point equation from Taylor approximation. The result is a linear equation in small signal values.

From the wind turbine models for the actuator, drive-train and generator in equations (3.3), (2.21) and (2.24), the following states exist:

- ***Actuator***

$$\dot{\beta}=\frac{-\beta+\beta_d}{\tau_\beta}$$

- ***Drive-train***

$$\dot{\theta}_{wt} = \omega_{wt}$$

$$\dot{\theta}_m = \omega_m$$

$$\dot{\omega}_{wt} = \frac{1}{J_T}(T_{wt} - T_{lss})$$

$$\dot{\omega}_m = \frac{1}{JG}(T_{hss} - T_e - b_d\omega_m)$$

where:

$$T_{lss} = K_s \theta_k + C_s \dot{\theta}_k = K_s(\theta_{wt} - \frac{\theta_m}{n_g}) + C_s(\omega_{wt} - \frac{\omega_m}{n_g}) = n_g T_{hss}$$

- ***Generator***

$$\dot{\psi}_{ds} = -R_s i_{ds} + \omega_s L_s i_{qs} + \omega_s L_m i_{qr} + v_{ds}$$
$$\dot{\psi}_{qs} = -\omega_s L_s i_{ds} - R_s i_{qs} - \omega_s L_m i_{dr} + v_{qs}$$
$$\dot{\psi}_{dr} = \omega_s L_m i_{qs} - R_r i_{dr} + \omega_s L_r i_{qr} - \omega_m L_m i_{qs} - \omega_m L_r i_{qr} + v_{dr}$$
$$\dot{\psi}_{qr} = -\omega_s L_m i_{ds} - \omega_s L_r i_{dr} - R_r i_{qr} + \omega_m L_m i_{ds} + \omega_m L_r i_{dr} + v_{qr}$$

The linear approximation for the wind turbine around an operating point is:

$$\delta\dot{\beta}(t) = \left.\frac{d\dot{\beta}}{d\beta}\right|_p \delta\beta(t) + \left.\frac{d\dot{\beta}}{d\beta_d}\right|_p \delta\beta_d(t)$$

$$\dot{\theta}_{wt}(t) = \left.\frac{d\dot{\theta}_{wt}}{d\omega_{wt}}\right|_{\omega_{wtp}} \delta\omega_{wt}(t)$$

$$\delta\dot{\theta}_m(t) = \left.\frac{d\dot{\theta}_m}{d\omega_m}\right|_{\omega_{mp}} \delta\omega_m(t)$$

$$\delta\dot{\omega}_{wt}(t) = \left.\frac{d\dot{\omega}_{wt}}{d\theta_{wt}}\right|_p \delta\theta_{wt}(t) + \left.\frac{d\dot{\omega}_{wt}}{d\theta_m}\right|_p \delta\theta_m(t) + \left.\frac{d\dot{\omega}_{wt}}{d\omega_{wt}}\right|_p \delta\omega_{wt}(t) + \left.\frac{d\dot{\omega}_{wt}}{d\omega_m}\right|_p \delta\omega_m(t) + \left.\frac{d\dot{\omega}_{wt}}{dT_{wt}}\right|_p \delta T_{wt}(t)$$

$$\delta\dot{\omega}_m(t) = \left.\frac{d\dot{\omega}_m}{d\theta_{wt}}\right|_p \delta\theta(t)_{wt} + \left.\frac{d\dot{\omega}_m}{d\theta_m}\right|_p \delta\theta_m(t) + \left.\frac{d\dot{\omega}_m}{d\omega_{wt}}\right|_p \delta\omega_{wt}(t) + \left.\frac{d\dot{\omega}_m}{d\omega_m}\right|_p \delta\omega_m(t) + \left.\frac{d\dot{\omega}_m}{dT_{wt}}\right|_p \delta T_e^c(t)$$

$$\delta\dot{\psi}_{ds}(t) = \left.\frac{d\dot{\psi}_{ds}}{di_{ds}}\right|_p \delta i_{ds}(t) + \left.\frac{d\dot{\psi}_{ds}}{di_{qs}}\right|_p \delta i_{qs}(t) + \left.\frac{d\dot{\psi}_{ds}}{di_{qr}}\right|_p \delta i_{qr}(t) + \left.\frac{d\dot{\psi}_{ds}}{dv_{ds}}\right|_p \delta v_{ds}(t)$$

$$\delta\dot{\psi}_{qs}(t) = \left.\frac{d\dot{\psi}_{qs}}{di_{ds}}\right|_p \delta i_{ds}(t) + \left.\frac{d\dot{\psi}_{qs}}{di_{qs}}\right|_p \delta i_{qs}(t) + \left.\frac{d\dot{\psi}_{qs}}{di_{qr}}\right|_p \delta i_{qr}(t) + \left.\frac{d\dot{\psi}_{qs}}{dv_{qs}}\right|_p \delta v_{qs}$$

$$\delta\dot{\psi}_{dr}(t) = \left.\frac{d\dot{\psi}_{dr}}{di_{qs}}\right|_p \delta i_{qs}(t) + \left.\frac{d\dot{\psi}_{dr}}{di_{dr}}\right|_p \delta i_{dr}(t) + \left.\frac{d\dot{\psi}_{dr}}{di_{qr}}\right|_p \delta i_{qr}(t) + \left.\frac{d\dot{\psi}_{dr}}{d\omega_m}\right|_p \delta\omega_m(t) + + \left.\frac{d\dot{\psi}_{dr}}{dv_{dr}}\right|_p \delta v_{dr}(t)$$

$$\delta\dot{\psi}_{qr}(t) = \left.\frac{d\dot{\psi}_{qr}}{di_{ds}}\right|_p \delta i_{ds}(t) + \left.\frac{d\dot{\psi}_{qr}}{di_{dr}}\right|_p \delta i_{dr}(t) + \left.\frac{d\dot{\psi}_{qr}}{di_{qr}}\right|_p \delta i_{qr}(t) + \left.\frac{d\dot{\psi}_{qr}}{d\omega_m}\right|_p \delta\omega_m(t) + \left.\frac{d\dot{\psi}_{qr}}{dv_{qr}}\right|_p \delta v_{qr}(t)$$

The states x and inputs u and output y are given by:

$$x = [\beta\ , \theta_{wt}, \theta_m, \omega_{wt}, \omega_m, i_{ds}, i_{qs}, i_{dr}, i_{qr}]^T$$

$$u = [\beta_d, T_{wt}, T_e^c, v_{ds}, v_{dr}, v_{dr}, v_{qr}]^T$$

$$y = [\beta, \omega_{wt}, \omega_m, T_e]^T$$

Then the state-space model can be written as:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

and the matrices A, B and C are given by:

$$A = \begin{bmatrix}
-\frac{1}{\tau_\beta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & \frac{-K_s}{J_T} & \frac{K_s}{J_T n_g} & \frac{-C_s}{J_T} & \frac{C_s}{J_T n_g} & 0 & 0 & 0 & 0 \\
0 & \frac{K_s}{J_G n_g} & \frac{-K_s}{J_G n_g^2} & \frac{C_s}{J_G n_g} & \frac{-C_s}{J_G n_g^2} - \frac{b_d}{J_G} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -R_s & \omega_s L_s & 0 & \omega_s L_m \\
0 & 0 & 0 & 0 & 0 & -\omega_s L_s & -R_s & -\omega_s L_m & 0 \\
0 & 0 & 0 & 0 & -L_m i_{qs} - L_r i_{qr} & 0 & L_m(\omega_s - \omega_m) & -R_r & L_r(\omega_s - \omega_m) \\
0 & 0 & 0 & 0 & L_m i_{ds} + L_r i_{dr} & -L_m(\omega_s - \omega_m) & 0 & -L_r(\omega_s - \omega_m) & -R_r
\end{bmatrix}$$

$$B=\begin{bmatrix} \frac{1}{\tau_\beta} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_T} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{J_G} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{n_p L_m V_s \sqrt{3}}{L_s \omega_s} \end{bmatrix}$$

- ***Operating point:***

Wind speed=10 m/s, generator speed ω_m=87.22 rad/s; generator currents are i_{ds}=53.2*A*, i_{qs}=*2217A,* i_{dr}=*-52.1A,* i_{qr}=*-2293A,* $\tau_\beta = 1$. Wind turbine parameters are presented in Table A.1.

By computing the observability matrix for state-space systems, for an *n*-by-*n* matrix *A* and a *p*-by-*n* matrix *C*, *obsv(A,C)* returns the observability matrix

$$ob=\begin{vmatrix} C \\ CA \\ CA^2 \\ . \\ . \\ CA^{n-1} \end{vmatrix}$$

where *n* and *p* are the dimensions of state and output, respectively.

Matlab commands were used to calculate the observability matrix *ob* and the number of unobservable states, *unob*:

```
ob = obsv(A,C);
unob = length(A)-rank(Ob)
```

The resulting number of unobservable states is five, meaning that the system is not completely observable. Therefore, the model can be revised by changing the drive-train model from 4th order to 3rd order and, then, substituting equation (2.23) into equation (2.21), giving the drive-train model as:

$$\begin{bmatrix} \dot{\theta}_{wt} \\ \dot{\theta}_{m} \\ \\ \dot{\omega}_{wt} \\ \dot{\omega}_{m} \end{bmatrix} = \begin{bmatrix} \omega_{wt} \\ \omega_{m} \\ \\ \frac{1}{J_T}(T_{wt} - T_{lss}) \\ \frac{1}{JG}(T_{hss} - T_e - b_d \omega_m) \end{bmatrix} \rightarrow \begin{bmatrix} \dot{\theta}_{k} \\ \\ \dot{\omega}_{wt} \\ \dot{\omega}_{m} \end{bmatrix} = \begin{bmatrix} \omega_{wt} - \frac{\omega_m}{n_g} \\ \\ \frac{1}{J_T}(T_{wt} - T_{lss}) \\ \frac{1}{JG}(T_{hss} - T_e - b\omega_m) \end{bmatrix} \tag{4.29}$$

The linear approximation for $\delta\dot{\theta}_k(t)$ around an operating point is:

$$\delta\dot{\theta}_k(t) = \left.\frac{d\dot{\theta}_k}{d\omega_{wt}}\right|_p \delta\omega_{wt}(t) + \left.\frac{d\dot{\theta}_k}{d\omega_m}\right|_p \delta\omega_m(t)$$

where:

$$x = [\beta \ , \theta_k , \omega_{wt} , \omega_m , i_{ds} , i_{qs} , i_{dr} , i_{qr}]^T$$

$$u = [\beta_d , T_{wt} , T_e^c , v_{ds} , v_{dr} , v_{dr} , v_{qr}]^T$$

$$y = [\beta , \omega_{wt} , \omega_m , T_e]^T$$

The matrices *A, B, C* and *D* are:

$$A = \begin{bmatrix} -\frac{1}{\tau_\beta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{-1}{n_g} & 0 & 0 & 0 & 0 \\ 0 & \frac{-K_s}{J_T} & \frac{-C_s}{J_T} & \frac{C_s}{J_T n_g} & 0 & 0 & 0 & 0 \\ 0 & \frac{K_s}{J_G n_g} & \frac{C_s}{J_G n_g} & \frac{-C_s}{J_G n_g^2} - \frac{b_d}{J_G} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -R_s & \omega_s L_s & 0 & \omega_s L_m \\ 0 & 0 & 0 & 0 & -\omega_s L_s & -R_s & -\omega_s L_m & 0 \\ 0 & 0 & 0 & -L_m i_{qs} - L_r i_{qr} & 0 & L_m(\omega_s - \omega_m) & -R_r & L_r(\omega_s - \omega_m) \\ 0 & 0 & 0 & L_m i_{ds} + L_r i_{dr} & -L_m(\omega_s - \omega_m) & 0 & -L_r(\omega_s - \omega_m) & -R_r \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{\tau_\beta} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_T} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{J_G} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{n_p L_m V_s \sqrt{3}}{L_s \omega_s} \end{bmatrix}, \quad D = \text{zeros}\,(4,7)$$

From the calculation of the observability matrix *ob* it follows that there are four unobservable states. Thus, the state-space model is still not observable. Therefore, the model requires further revision.

From the voltages, currents and flux linkages are expressed by the d-axis and q-axis components in Section 2.6.1 (Chapter 2). The stator voltage vector was oriented to the q-axis of the reference frame. Therefore, it should be noted that the stator voltage vector and the order of the DFIG model can be modified. From equation (2.55), the model for the generator can be written as:

$$\begin{aligned}\sigma L_r \frac{di_{dr}}{dt} &= -R_r i_{dr} + (\omega_s - \omega_m)\sigma L_r i_{qr} + v_{dr} \\ \sigma L_r \frac{di_{qr}}{dt} &= -R_r i_{qr} - (\omega_s - \omega_m)(\sigma L_r i_{dr} + \frac{\sqrt{3} L_m V_s}{\omega_s L_s}) + v_{qr}\end{aligned} \tag{4.30}$$

From the models for actuator, drive-train and generator in equations (3.3), (4.29) and (4.30), the linear state-space model for the wind turbine around an operating point is as follows:

The states, inputs and outputs are defined as:

$$x = [\beta \ , \theta_k , \omega_{wt} , \omega_m , i_{dr} , i_{qr}]^T$$

$$u = [\beta_d , T_{wt} , T_e^c , v_{dr} , v_{qr}]^T$$

$$y = [\beta , \omega_{wt} , \omega_m , T_e]^T$$

A, B, C and *D* are:

$$A = \begin{bmatrix} -\frac{1}{\tau_\beta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{-1}{n_g} & 0 & 0 \\ 0 & -\frac{K_S}{J_T} & \frac{-C_S}{J_T} & \frac{C_S}{J_T n_g} & 0 & 0 \\ 0 & \frac{K_S}{J_G n_g} & \frac{C_S}{J_G n_g} & \frac{-C_S}{J_G n_g^2} & 0 & 0 \\ 0 & 0 & 0 & -i_{qr} & \frac{-R_r}{\sigma Lr} & (\omega_S - \omega_m) \\ 0 & 0 & 0 & i_{dr} + \frac{L_m V_s \sqrt{3}}{L_s \omega_s \sigma L_r} & -(\omega_S - \omega_m) & \frac{-R_r}{\sigma L_r} \end{bmatrix}$$

$$B=\begin{bmatrix} \frac{1}{\tau_\beta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_T} & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{J_G} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},\quad C=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{0.8383 n_p L_m V_s \sqrt{3}}{L_s \omega_s}\sigma L_r \end{bmatrix},\quad \mathrm{D}=\mathrm{zeros}\ (4,5)$$

The number of unobservable states is zero and, consequently, the state-space model is observable.

4.3.2 Linear model validation

Comparing the output of the non-linear model with the output from the state-space model at a wind speed 10 m/s, as shown in Figures (4.4) to (4.8), the outputs from both models are very similar.

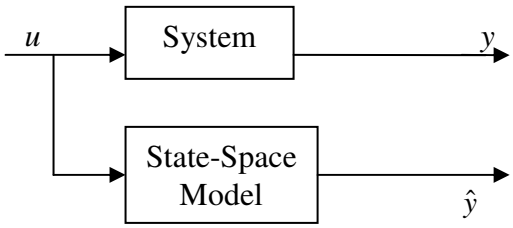


Figure 4.3: Comparing non-linear model and state-space model

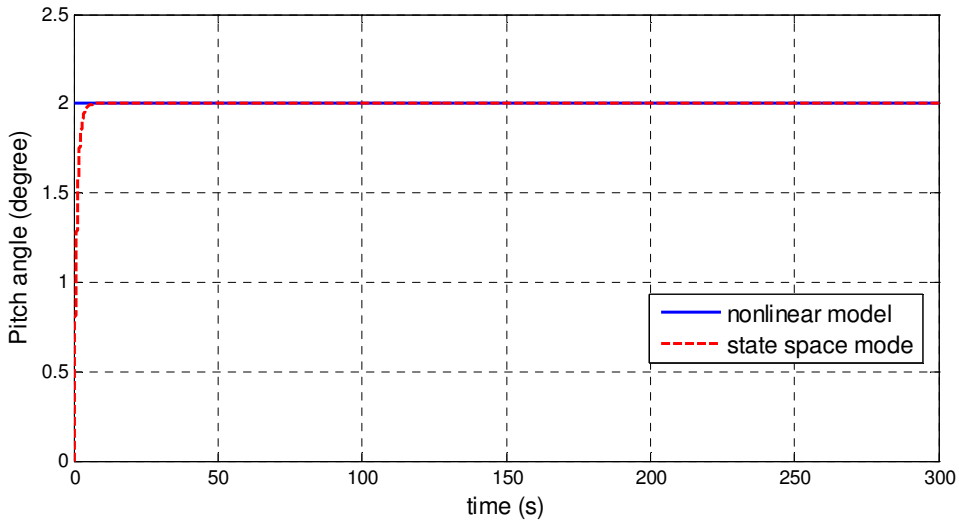


Figure 4.4: Non-linear model and state-space pitch angle outputs. Here, the wind speed (10 m/s) is below the rated value, therefore the pitch angle controller is inactive.

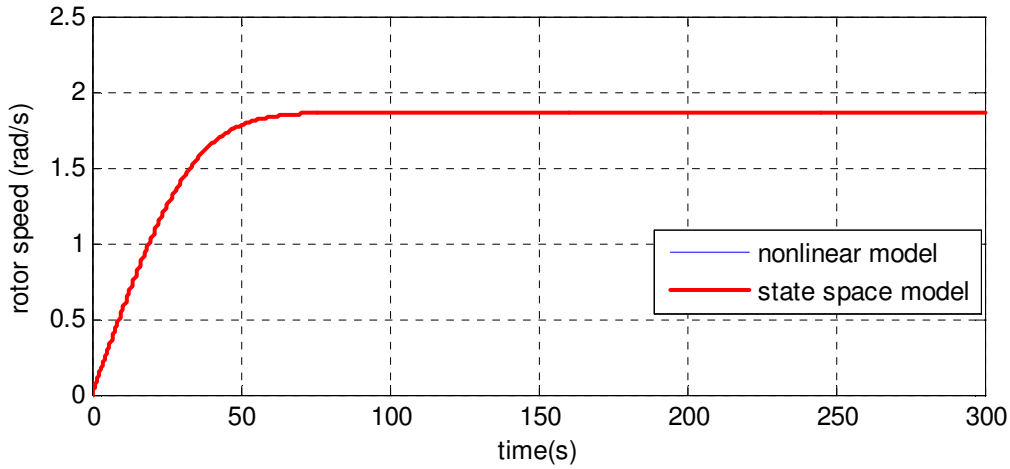


Figure 4.5: Non-linear model and state-space rotor speed outputs

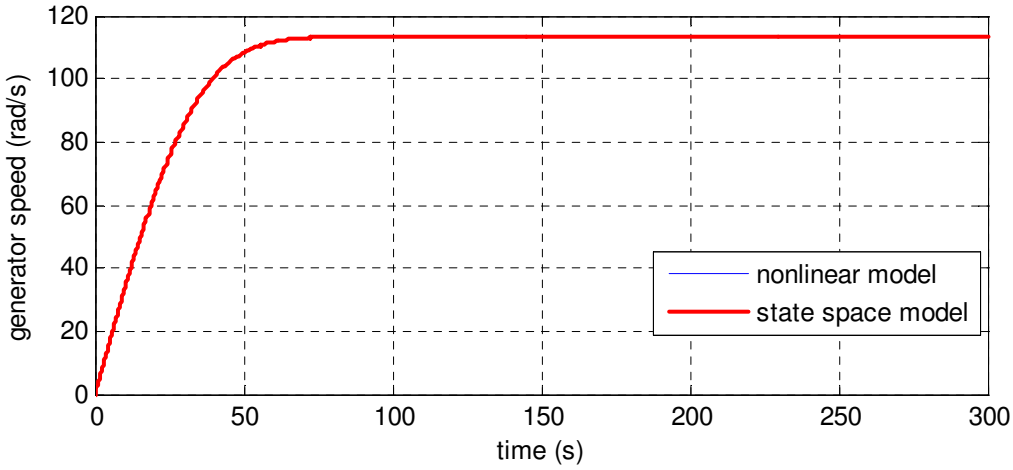


Figure 4.6: Non-linear model and state-space generator speed outputs

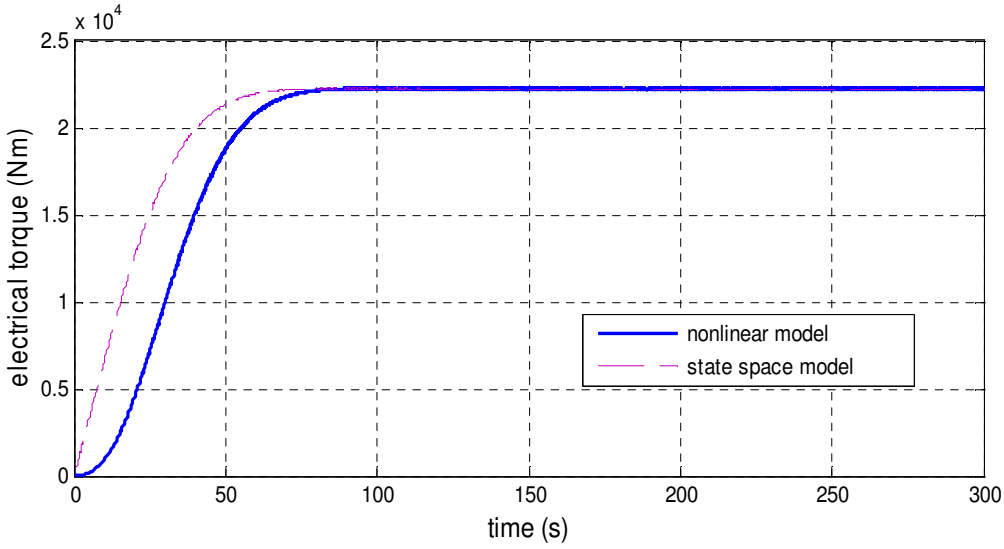


Figure 4.7: Non-linear model and state-space electrical torque outputs

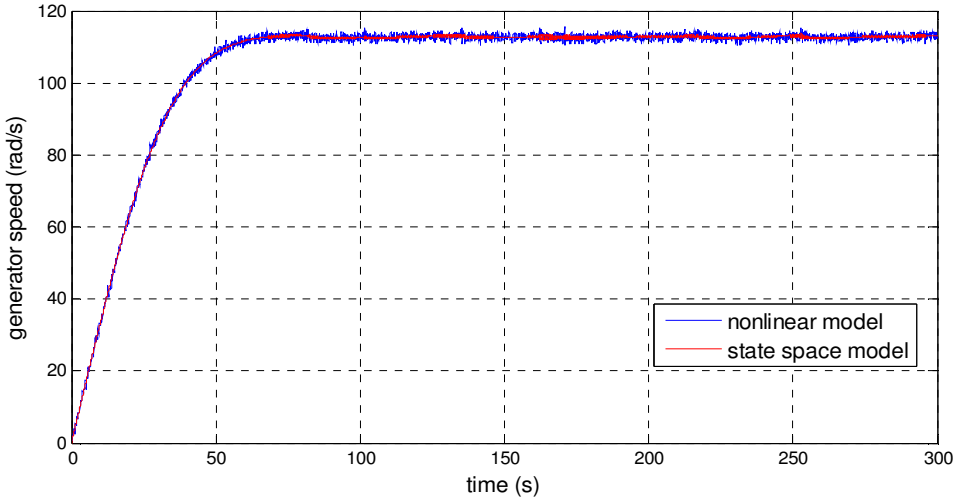


Figure 4.8: Non-linear model and state-space generator speed outputs, where drive-train disturbance covariance is 0.1 and the sensor noise covariance is 0.5

4.3.3 Fault detection and isolation

One approach for fault isolation is to design a set of structured residuals. Each residual is designed to be sensitive to a particular type of fault and insensitive to other types of fault (Gertler, 1991) [86]. To implement this approach, two steps should be taken. The first step is to define the sensitive and insensitive relationships between residuals and faults for each type of fault, represented as mathematical equations. The second step is to design a set of residual generations according to those equations. Faults are classified into three groups: actuator, sensor and component (rotor, drive-train and generator) faults. The system equation can be expressed as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + R_1^a f^a(t) + R_1^s f^s(t) + R_1^c f^c(t) + d(t) \\ y(t) &= Cx(t) + Du(t) + R_2^a f^a(t) + R_2^s f^s(t) + R_2^c f^c(t) \end{aligned} \tag{4.31}$$

where R_1 and R_2 are fault matrices and a, s and c denote actuator, sensor and component, respectively.

According to Chiang et al. (2001) [87], the sensor, actuator and component fault matrices can be represented by equations (4.32) and (4.33).

$$R_1 = \begin{cases} 0 & sensor\ fault \\ B & actuator\ fault \\ I & component\ fault \end{cases} \tag{4.32}$$

$$R_2 = \begin{cases} I & sensor\ fault \\ D & actuator\ fault \\ 0 & component\ fault \end{cases} \tag{4.33}$$

Substituting equations (4.32) and (4.33) into equation (4.31), gives the following system equation:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Bf^a(t) + If^c(t) + d(t) \\ y(t) &= Cx(t) + Du(t) + Df^a(t) + If^s(t) \end{aligned} \tag{4.34}$$

Referring to the performance indices of residual generation in equations (4.20) and (4.28), the performance indices can be rewritten as:

$$J_{af}(K,Q) = \sup_{\omega\in[\omega_1,\omega_2]} \bar{\sigma}\{-[Q_2D + QC(j\omega I - A + KC)^{-1}(B - KD)]\ \} \tag{4.35}$$

$$J_{sf}(K,Q) = \sup_{\omega\in[\omega_1,\omega_2]} \bar{\sigma}\{[QI + QC(j\omega I - A + KC)^{-1}(-KI)]^{-1}\} \tag{4.36}$$

$$J_{cf}(K,Q) = \sup_{\omega\in[\omega_1,\omega_2]} \bar{\sigma}\{-[QC(j\omega I - A + KC)^{-1})]\} \tag{4.37}$$

$$J_d(K) = \| (A - KC)^{-1} \| \tag{4.38}$$

From the above equations, the performance indices of the actuator $J_{af}(K,Q)$, sensor $J_{sf}(K,Q)$, component faults $J_{cf}(K,Q)$ and disturbance effects $J_d(K)$ need to be optimised as follows:

- The effect of the actuator, sensor and component faults can be maximised via minimisation of $J_{af}(K,Q)$, $J_{sf}(K,Q)$ and $J_{cf}(K,Q)$ indices.

- Disturbance effects in the steady state period on the residual can be minimised by minimising $J_d(K)$ index.

4.3.4 Multi-objective optimisation using the method of multi-objective genetic algorithm

The problem can now be stated as minimising the criteria in equations (4.35), (4.36), (4.37) and (4.38). This is a multi-objective optimisation problem and, hence, the use of a genetic algorithm (GA) is proposed to solve the problem.

A multi-objective genetic algorithm (MOGA) is used to minimise the objective functions (see Appendix C). This is more suitable than other approaches such as conventional genetic algorithms because it is not necessary to apply an equality constraint with a MOGA [88]. The method finds the solution to problems with two or more objectives to be satisfied together. Often, such objectives are in conflict with each other, and are expressed in different units. Because of their nature, multi-objective optimization problems normally have not one but a set of solutions, known as Pareto points or Pareto optimal solutions [89].

4.3.4.1 Minimisation of two objective functions for sensor faults

Firstly, two objective functions, $Jsf(K,Q)$ and $Jd(K)$, with $n+ p^2$ decision variables should be minimised. Mathematically, the problem can be written as:

Define:

$F(X) = [F_1(X); F_2(X)]$

where $F_1(X)= J_{sf}(K,Q)$ *and* $F_2(x)=J_d(K)$, and $X=\{x_1,\ldots\ldots, x_{n+p^2}\}$ is a vector of decision variables.

The problem is to minimise $F(X)$ subject to:

- $F_1(X) \leq 0$ *and* $F_2(X) = 0$.
- $F_1(X)$: inequality constraint evaluated at X

- $F_2(X)$: equality constraint evaluated at X

In the vector function $F(x)$, some of the objectives may be in conflict with others and some have to be minimised while others are maximised. The constraints define the feasible region X, and any point $x \in X$ is a feasible solution. There is rarely a situation in which all $F(X)$ have an optimum in X at a common point. Therefore, in the absence of preference information, solutions to multi-objective problems are compared using the notion of Pareto dominance.

Without loss of generality, in a minimisation problem for all objectives, a solution X_1 dominates a solution X_2 if the two following conditions are true:

- X_1 is no worse than X_2 for all objectives, i.e., $f_i(X_1) \leq f_i(X_2)$
- X_1 is strictly better than X_2 for at least one objective, i.e., $f_i(X_1) < f_i(X_2)$.

Then, a solution is said to be Pareto-optimal if it is not dominated by any other possible solution, as described above. Thus, the Pareto-optimal solutions to a multi-objective optimisation problem form the Pareto front or Pareto-optimal set [55].

The performance indices $J_{sf}(K,Q)$ and $J_d(K)$ are functions in K and Q. Therefore, the parameters set to be designed are the observer gain matrix and residual weighting factor matrix. The matrix K must achieve the stability of the observer and optimisation of the performance indices. Ackermann's formula is used to parameterise the matrix K [85]:

$$K_s = \left[A^n + \alpha_1 A^{n-1} + \alpha_2 A^{n-2} + \cdots + \alpha_{n-1} A + \alpha_n I \right] \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

where the desired eigenvalues are:

$$(s - p_1)(s - p_2) \cdots (s - p_n) = s^n + \alpha_1 s^{n-1} + \alpha_2 s^{n-2} + \cdots + \alpha_{n-1} s + \alpha_n$$

To improve the design, the desired eigenvalues are assigned in pre-defined regions to meet stability and response requirements, as in the following equations [46]:

$$p_i = L_i + (U_i - L_i)\sin^2(x_i),\ i = 1,...,n$$

$$L_i \leq p_i \leq U_i$$

where:

U_i=[-6 -10 -1 -3 -8.5 -14]; L_i=[-8 -12 -2 -4 -9.5 -20];

where U_i and L_i are the lower and higher limits for the eigenvalues, respectively; index x_i can be freely selected. Constrained performance indices have now been transformed into unconstrained indices, as a function of X:

$$K_s = f\left(x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6\right)$$

$$Q_s = \begin{bmatrix} x_7 & x_{11} & x_{15} & x_{19} \\ x_8 & x_{12} & x_{16} & x_{20} \\ x_9 & x_{13} & x_{17} & x_{21} \\ x_{10} & x_{14} & x_{18} & x_{22} \end{bmatrix}$$

where $x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6$ are the desired eigenvalues (p_i), K_s is the observer gain for sensor faults and Q_s is the residual weighting factor matrix for the sensor faults.

4.3.4.2 Multi-objective genetic algorithm procedure

This section summarises the steps of the MOGA procedure using the Matlab MOGA Toolbox for solving the performance indices in equations (4.35) - (4.38). Firstly, two objective functions are minimised for the sensor faults ($J_{sf}(K,Q)$ and $J_d(K)$), as shown in Figure 4.9.

Step 1:

Define the wind turbine state-space model.

Step 2:

Describe the fitness function ($F(X)$) as a multi-objective vector function ($[F_1(X); F_2(X)]$) and input it to the Matlab MOGA Toolbox.

Step 3:

Define the following MOGA parameters:

1) Number of independent variables = 22 for the fitness function.
2) Initial range of variables, the fitness function lower and upper bounds for the entries of the vectors in the initial population. The initial range can be specified as a matrix with two rows and initial length columns. The first row contains the lower bounds for the entries of the vectors in the initial population, while the second row contains the upper bounds.
3) Population size (how many individuals are in each generation).
4) Number of generations, which specifies the maximum number of iterations the GA performs.

The tuning parameters were set as follows: initial range of variables= [0; 10000], population size: 75, number of generations: 200.

a) Step 4:

Generate an initial randomly chosen population that satisfies the bounds.

b) Step 5:

Evaluate all objective functions using the distance measure function:

$$F_1^{\,j}(X) \cap F^{k}{}_2(X) < F_1^{\,j-1}(X) \cap F^{k-1}{}_2(X)$$

in order to determine the concentration of the population.

c) Step 6:

Plot various aspects of the GA as:

1) Score histogram of the parents. This shows which parents contribute to each generation.
2) Pareto front plots, which show values for all best possible solutions.

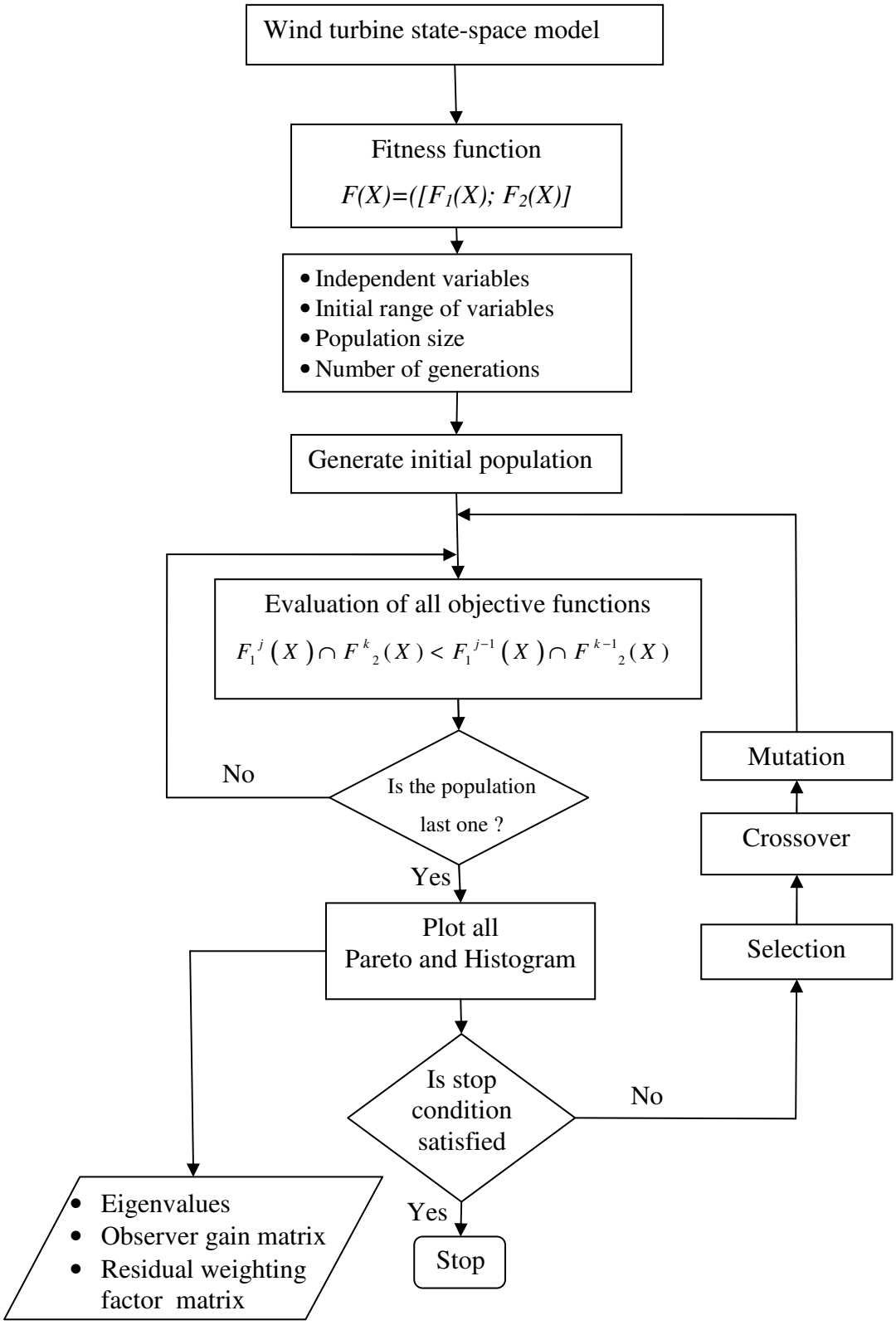


Figure 4.9: Flow chart of the multi-objective simulation optimisation framework

d) Step 7:

Stopping criteria determine what causes the algorithm to end:

1) Generations specify the maximum number of iterations the GA performs.
2) Fitness limit: if the best fitness value is less than or equal to the value of fitness limit, then the algorithm stops.

e) Step 8:

1) Selection chooses parents for the next generation based on their scaled values from the fitness functions.
2) Crossover combines two individuals, or parents, to form a new individual, or child, for the next generation.
3) Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the GA to search a broader space.

f) Step 9:

Select the best values from Pareto front to determine the eigenvalues, the observer gain matrix and the residual weighting factor matrix.

The top chart in Figure 4.10 illustrates the score diversity for each objective. The bottom chart in Figure 4.10 shows the Pareto front, which plots the Pareto front for every generation. From Figure 4.10 it can be seen that there is only one optimal value on the Pareto front figure, that gives $J_{sf}(K,Q)$=0, which means the maximal performance index of the sensor is infinity and $J_d(K)$ = 258.

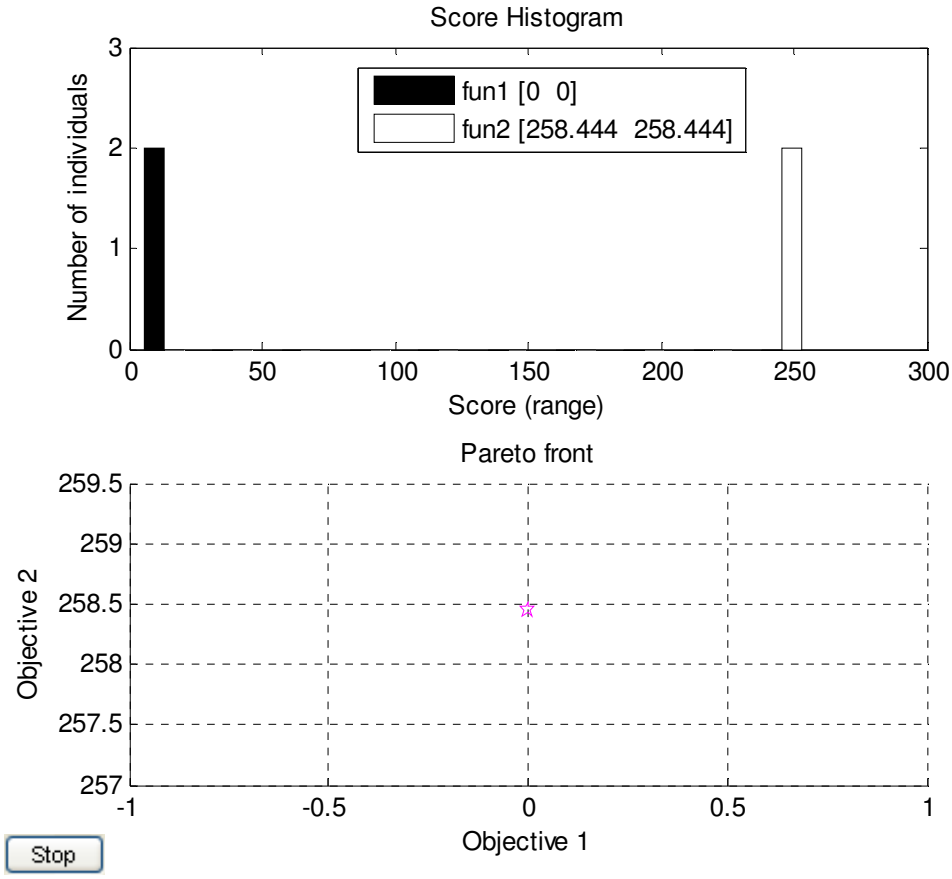


Figure 4.10: Number of generations and the Pareto front for sensor index (fun1 and objective 1 represent $J_{sf}(K,Q)$, fun2 and objective 2 represent $J_d(K)$)

From the results shown in Figure 4.10, the eigenvalues at these indices P_s, the matrix K_s and residual weighting factor matrix Q_s are:

$$P_s = \begin{bmatrix} -8 & -11.7 & -1 & -4 & -8.6 & -20 \end{bmatrix}$$

$$K_s = \begin{bmatrix} 0.012 & -7.6\text{x}10^{-06} & 3.7\text{x}10^{-08} & 7.3\text{x}10^{-10} \\ 0.0001 & -2 & -0.017 & -1.8\text{x}10^{-14} \\ -0.0003 & 15.26 & -0.016 & 3.6\text{x}10^{-14} \\ 0.182 & -4338.1 & 17.35 & -2.2\text{x}10^{-11} \\ -2.2\text{x}10^{-09} & 1.2\text{x}10^{-15} & -672 & -0.40 \\ -2.2\text{x}10^{-09} & 1.4\text{x}10^{-15} & 11.03 & -7.32 \end{bmatrix}$$

$$Q_s = \begin{bmatrix} 4426 & 4495 & 7814 & 5446 \\ 3015 & 4094 & 6897 & 1503 \\ 3836 & 3872 & 4308 & 8549 \\ 5144 & 6240 & 6211 & 5673 \end{bmatrix}$$

4.3.4.3 Minimisation of two objective functions for actuator faults

Optimisation of $J_{af}(K,Q)$ and $J_d(K)$ uses the same steps as in the first stage for the sensor faults. From the results in Figure 4.11, the best values of $J_{af}(K,Q)$ =-11681x10^4 and $J_d(K)$=258 are selected. The eigenvalues at these indices (P_a), the matrix K_a, and the residual weighting factor matrix (Q_a) are:

$$P_a = \begin{bmatrix} -7.4 & -11 & -1.14 & -4 & -8.5 & -20 \end{bmatrix}$$

$$K_a = \begin{bmatrix} 0.143 & 8.9\text{x}10^{-06} & -2.8\text{x}10^{-08} & 2.7\text{x}10^{-10} \\ -9.7\text{x}10^{-05} & -1.9144 & -0.0168 & 6.13\text{x}10^{-15} \\ 2.792\text{x}10^{-04} & 14.975 & -0.0165 & -1.2\text{x}10^{-14} \\ -0.1591 & -4228 & 17.536 & 7.8\text{x}10^{-12} \\ -9.5\text{x}10^{-10} & -1.8\text{x}10^{-16} & -672 & 0.0792 \\ -9.8\text{x}10^{-10} & -2.5\text{x}10^{-16} & 11.032 & -6.8542 \end{bmatrix}$$

$$Q_a = \begin{bmatrix} 8934 & 5713 & 2991 & 4010 \\ 7069 & 3547 & 2338 & 5786 \\ 7870 & 6083 & 5323 & 7954 \\ 5232 & 5901 & 2359 & 1312 \end{bmatrix}$$

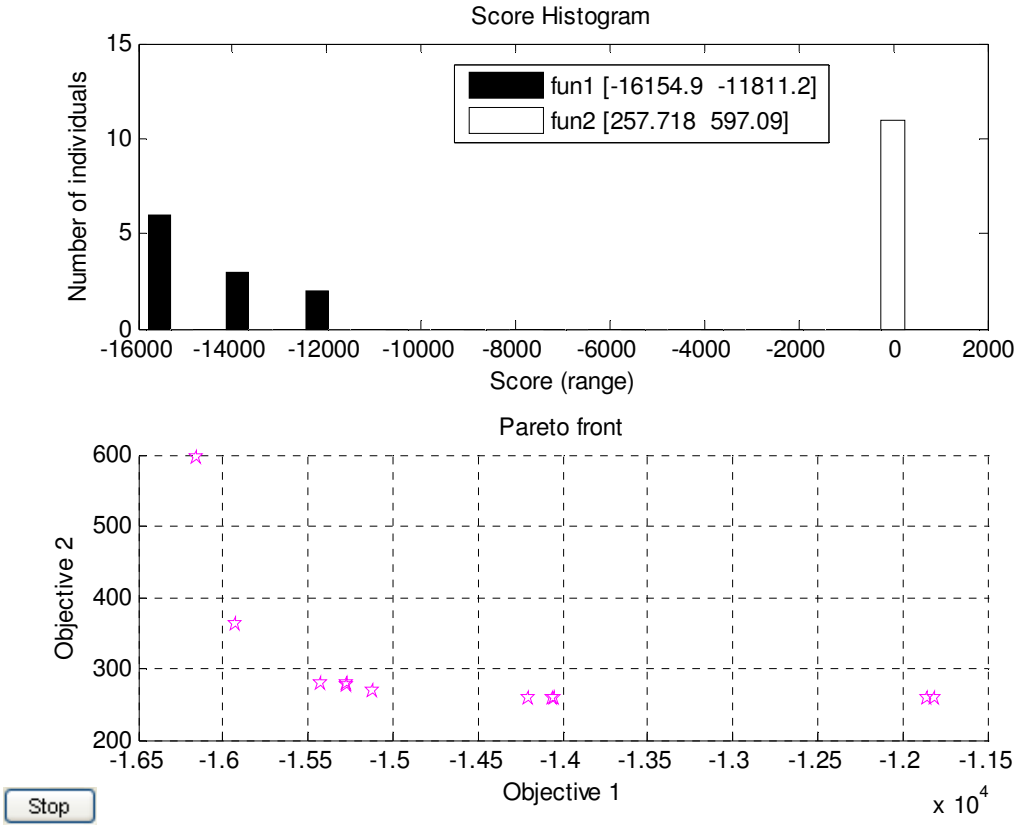


Figure 4.11: Number of generations and the Pareto front for actuator index (fun1 and objective 1 represent $J_{af}(K,Q)$, fun2 and objective 2 represent $J_d(K)$)

4.3.4.4 Minimisation of two objective functions for component faults

By repeating the same steps as in the first stage for the sensor faults, Figure 4.12 gives $J_{cf}(K,Q)$ =-3.2x10^6 and $J_d(K)$=258. The eigenvalues at these indices (P_c), the matrix K_c, and the residual weighting factor matrix (Q_c) are:

$$P_c = \begin{bmatrix} -8 & -11.5 & -1.7 & -4 & -8.5 & -20 \end{bmatrix}$$

$$K_c = \begin{bmatrix} 0.143 & 8.9\text{x}10^{-06} & -2.8\text{x}10^{-08} & 2.7\text{x}10^{-10} \\ -9.7\text{x}10^{-05} & -1.914 & -0.017 & 6.1\text{x}10^{-15} \\ 2.8\ \text{x}10^{-04} & 14.975 & -0.0165 & -1.3\text{x}10^{-14} \\ -0.159 & -4228 & 17.53 & 7.7\text{e-}12 \\ -9.4\text{x}10^{-10} & -1.8\text{x}10^{-16} & -672 & 0.0792 \\ -9.7\text{x}10^{-10} & -2.4\text{x}10^{-16} & 11.03 & -6.854 \end{bmatrix}$$

$$Q_c = \begin{bmatrix} 6078 & 3432 & 7310 & 7003 \\ 6248 & 4865 & 7088. & 4109 \\ 4834 & 6247 & 8685 & 5182 \\ 5562 & 3295 & 7117 & 1931 \end{bmatrix}$$

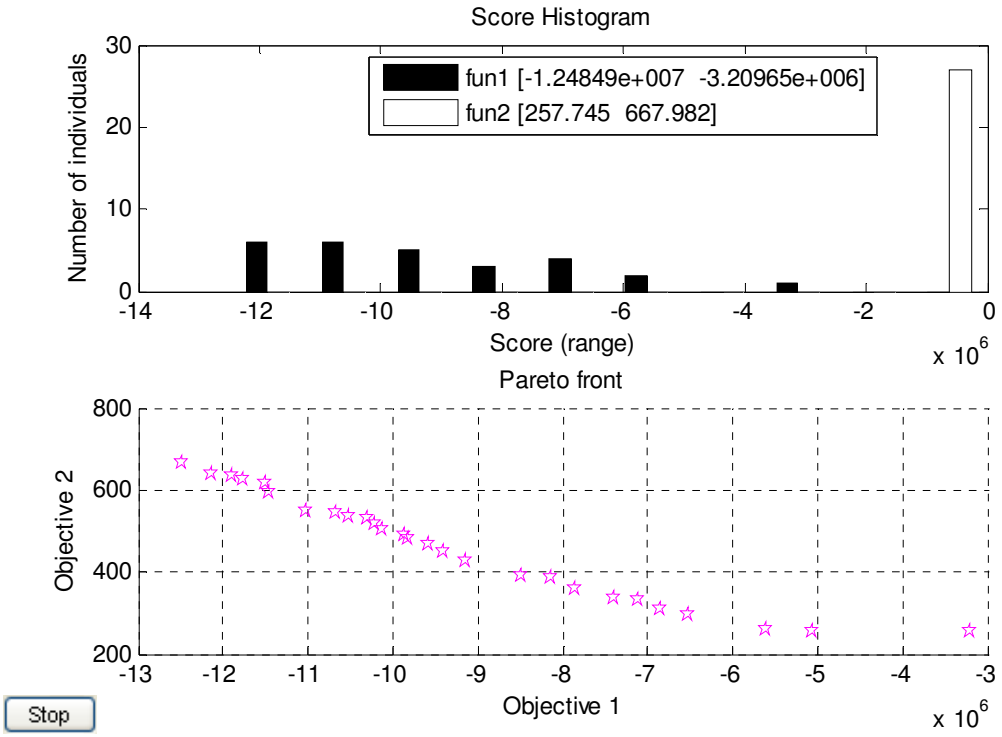


Figure 4.12: Number of generations and the Pareto front for the component index (fun1 and objective 1 represent $J_{cf}(K,Q)$, fun2 and objective 2 represent $J_d(K)$)

4.3.5 Observer-based FDI scheme

An observer-based residual generator is designed for the assessment of the performance of this method in wind turbines (see Figure 4.13). In this investigation different types of faults are applied: sensor, actuator or component fault. The aim is to classify the type of the fault in single fault cases, in other words to classify the single fault as being a sensor, actuator or component fault. Therefore, only one fault is assumed to occur in each simulation.

- Actuator fault: 10% change in the control voltage v_{qr}^{c} at 200 s:

$$v_{qr\_f}^{c}(t) = v_{qr}^{c}(t) + \Delta v_{qr}^{c}(t) \tag{4.39}$$

- Sensor fault: sensor of generator speed has been perturbed by 2% at 200 s:

$$\omega_{m\_f} = \omega_m + \Delta\omega_m \tag{4.40}$$

- Component fault: for example, dirt on blades or slight damaged blade causes 2% lower rotor torque due to changed aerodynamical efficiency.

$$T_{wt\_f} = T_{wt} + \Delta T_{wt} \tag{4.41}$$

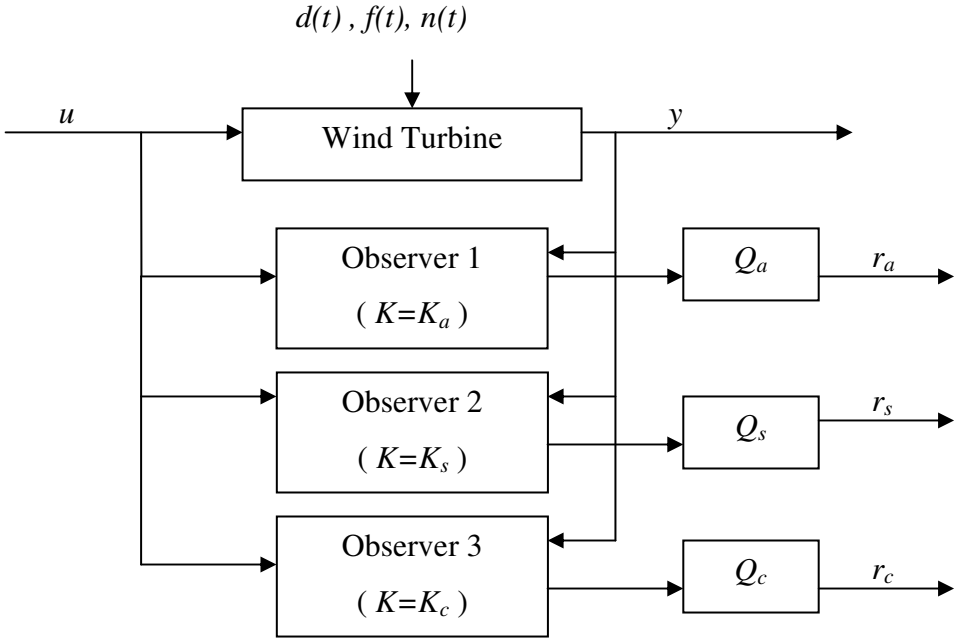


Figure 4.13: Observer-based residual generators using three observers designed to generate actuator, sensor and component fault residuals. r_a, r_s and r_c are actuator, sensor and component fault residuals, respectively.

4.3.6 Simulation results for observer-based FDI scheme

Figure 4.14 shows residuals for normal operation, when the wind turbine has only disturbance and noise signals.

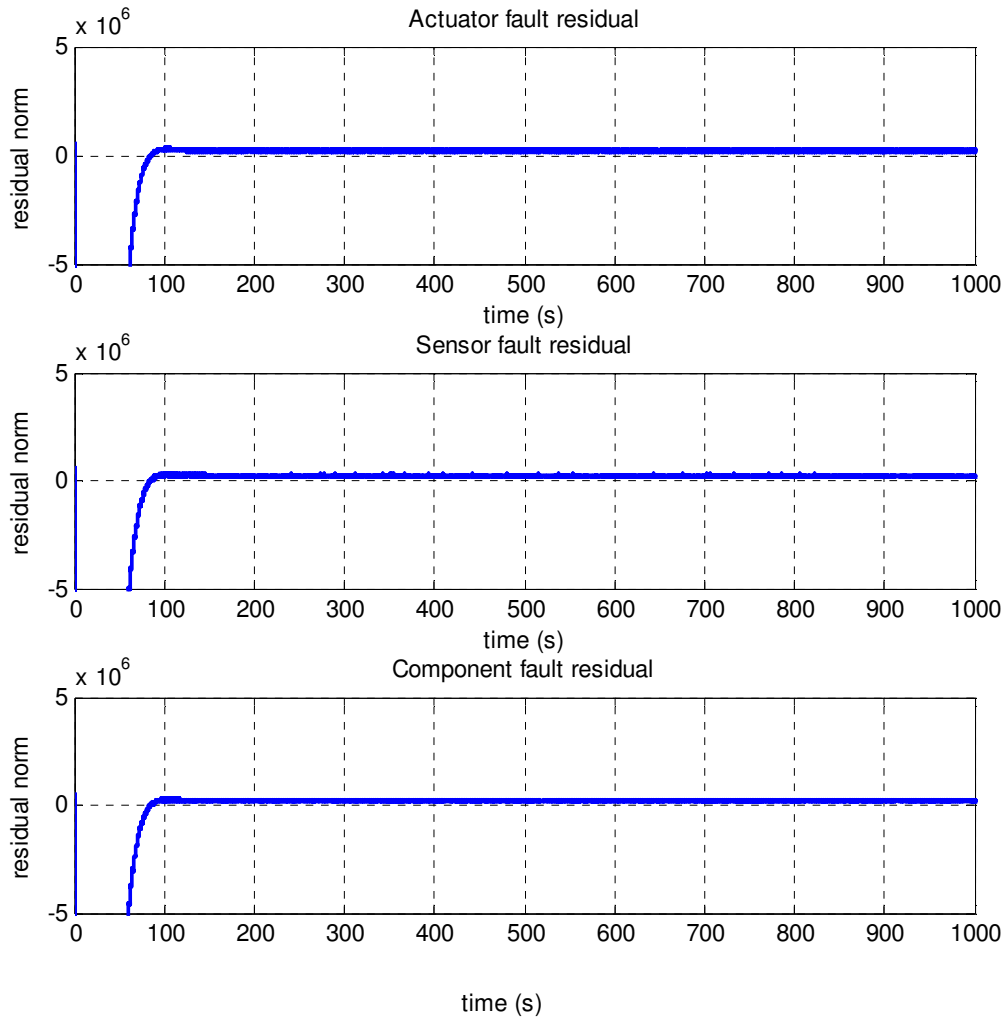


Figure 4.14: The residual norm when the wind turbine has disturbance and noise

Figure 4.15 shows when the actuator fault occurs; the residual fault appears for a short time, due to control loops bringing the variables back to their set point, even though the fault continues to be present in the system. This type of system behaviour makes an actuator fault difficult to diagnose; all residual generators are affected as a result of changing the output.

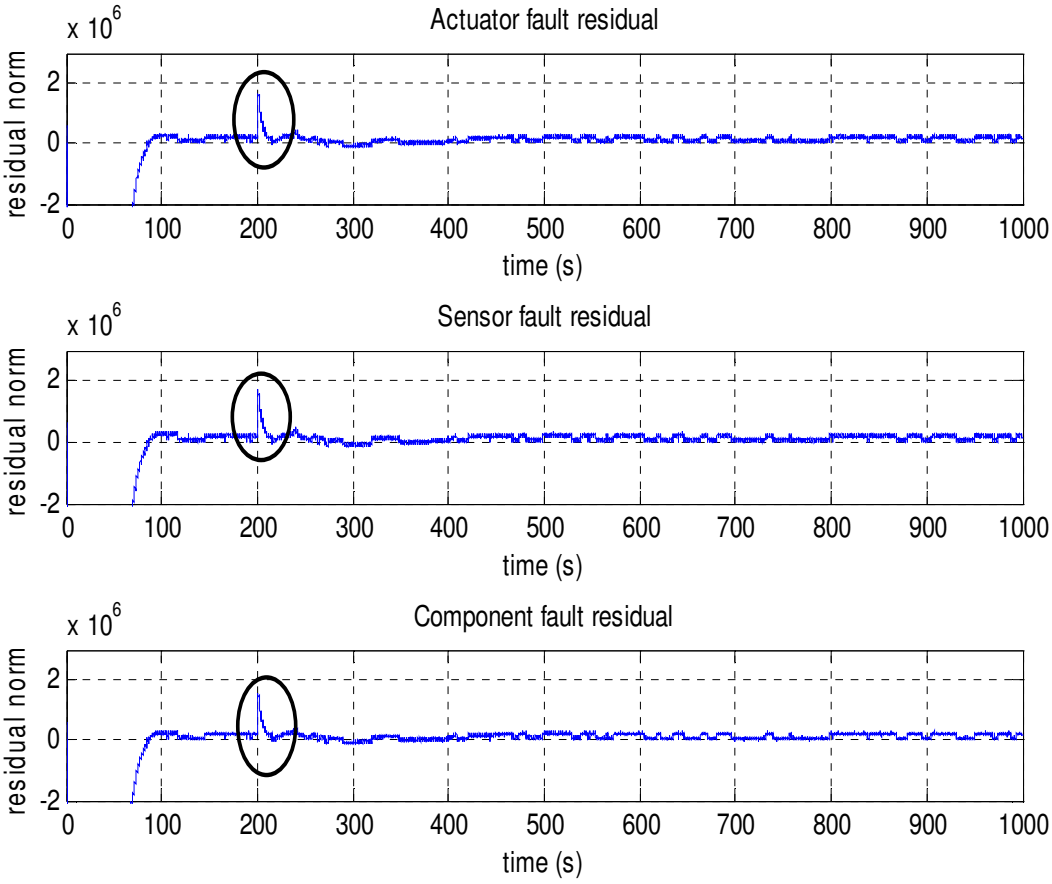


Figure 4.15: The residual norm, when the wind turbine has disturbance and noise as well as an actuator fault

When the sensor fault occurs (see Figure 4.16) there is an actuator fault, as a result of the measurement signal is used as a feedback signal to the controller. Figure 4.17 shows when the component fault occurs; here all fault residuals are affected (the faults occur at 200 s).

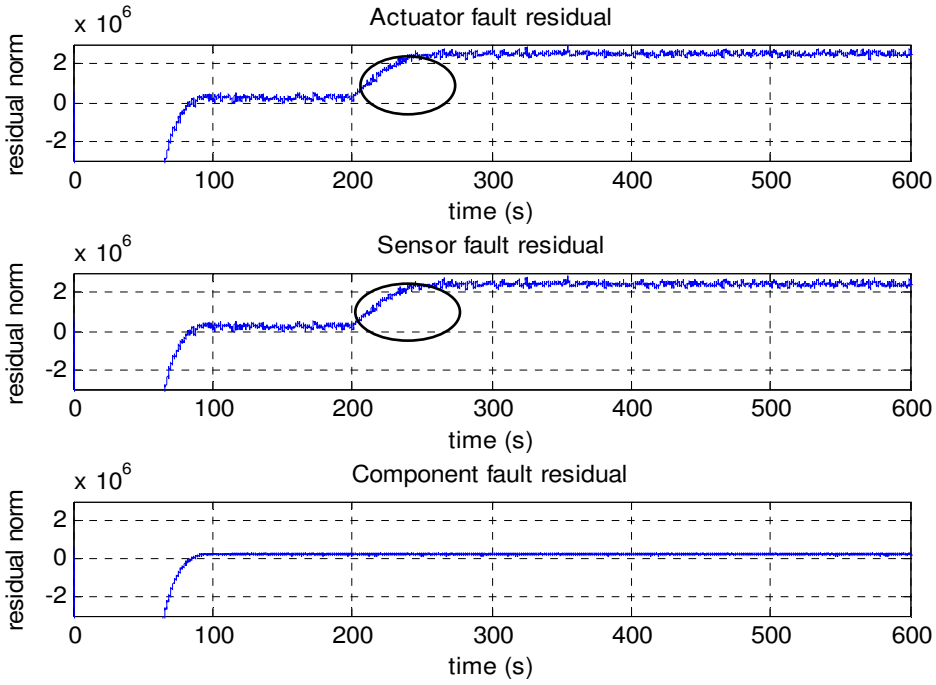


Figure 4.16: The residual norm, when the wind turbine has disturbance and noise as well as a sensor fault

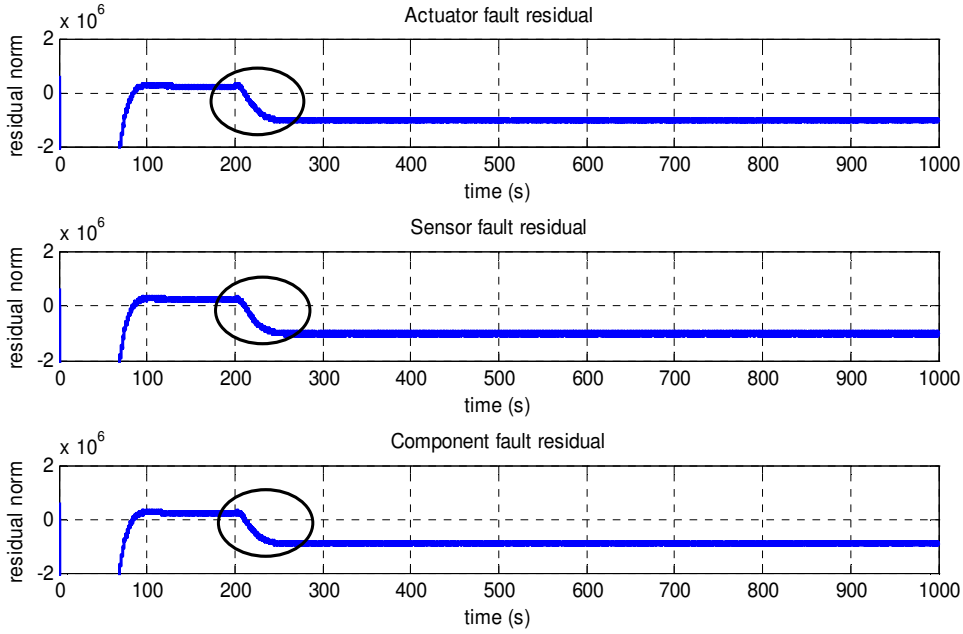


Figure 4.17: The residual norm, when the wind turbine has disturbance and noise as well as a component fault

4.4 Observer-based sensor fault detection and isolation scheme

A successful FDI should be accompanied by a fault isolation procedure to isolate a particular fault from others; for example, to determine in which sensor, actuator or component the fault happened. The observer-based residual generator approach is suitable for detecting a fault [52], but to isolate the fault, a new method is proposed here.

4.4.1 Sensor fault detection and isolation scheme

To design a robust observer-based sensor FDI, it was assumed that only one sensor fault occurs and that all actuators and components are fault free. Then, from equations (4.31) - (4.33), the system equation can be expressed as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + d(t) \\ y(t) &= Cx(t) + R_2 f(t) \end{aligned} \tag{4.42}$$

Then, the residual generator can be created for each sensor as:

$$r_k(t) = Q_k[(Ce(t) + R_2 f(t))(C_m - C_k)] \tag{4.43}$$

where k is the number of the measurement sensor, $C_k \in R^{mxm}$ is obtained from the matrix C by deleting zero columns and assuming the k_{th} row is equal to zero. C_m is the matrix C without zero columns. From equation (4.43), it is obvious that each residual generator is driven, making all other residuals equal to zero.

The observer-based sensor FDI scheme is designed as shown in Figure 4.18. Each sensor residual (r_k) is separated from the output of the residual *(r)* by $r\,x\,(C_m - C_k)$, and, then, the dimension of r_k is modified using Q_k. The advantage of this approach is that it uses only one observer; alternative approaches use a bank of observers such as a structured residual set designed by a dedicated or a generalised observer scheme [91].

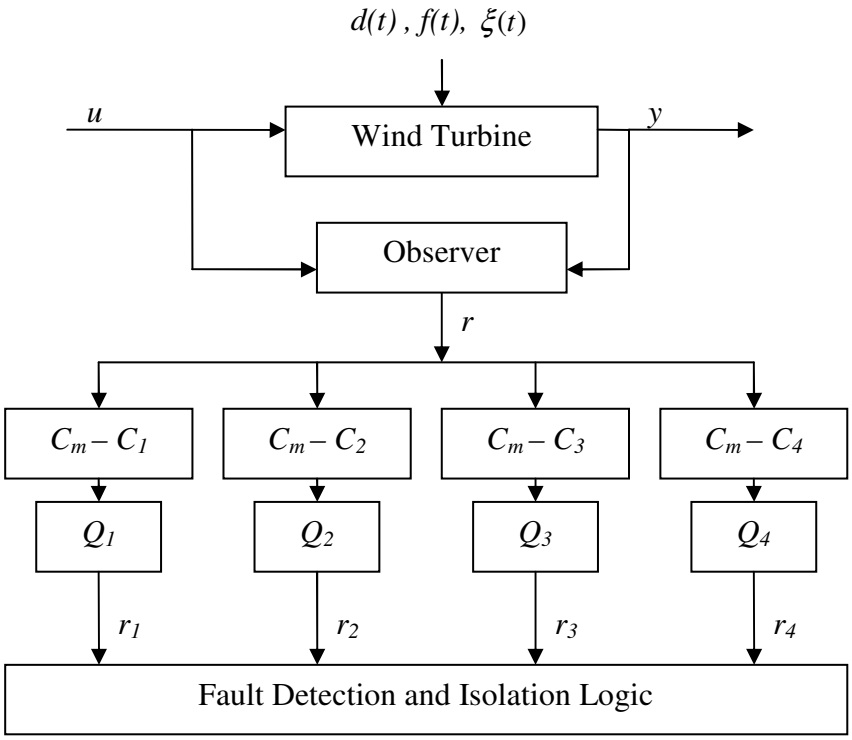


Figure 4.18: Sensor fault detection and isolation scheme

For simulation, the values of the residual weighting factors are selected as follows:

$$\frac{Q_1}{100} = \frac{Q_2}{1000} = Q_3 = 10 \;\; Q_4 = Q_s$$

Q_s are obtained using the MOGA method. Q_1, Q_2, Q_3 and Q_4 are residual weighting factors for pitch angle, rotor speed, and generator speed and torque sensors, respectively.

4.4.2 Simulation results for sensor fault detection isolation scheme

Simulation results are shown in Figure 4.19 to Figure 4.22. The faults are applied by multiplying the sensor signal by 1.05, i.e. an increase of 5% at 200 s, and the transient period is neglected. Figure 4.19 shows that the residual norm of the pitch sensor increases when there is a fault. Figure 4.20 shows that a fault has occurred in the rotor speed sensor; the faults are detected by the residual norm of the rotor and

generator rotational speed sensors. Figure 4.21 shows the case where a fault has occurred in the generator speed sensor and Figure 4.22 shows the case where a fault has occurred in the generator torque sensor. The speed of fault detection is very fast. Consequently, from these figures, the Boolean decision table can be constructed as shown in Table 4.1. If a fault occurs, the results can be compared with this fault signature table and the location of the fault can be determined, as shown in Figure 4.18. Therefore, fault detection and isolation are achieved.

Table 4.1: Boolean decision for sensor faults

Fault	Residual			
	Pitch residual	Rotor speed residual	Generator speed residual	Electrical torque residual
Pitch fault	1	0	0	0
Rotor speed fault	0	1	1	0
Generator speed fault	0	1	1	1
Electrical torque fault	0	0	0	1

Decision logic is the last stage in residual evaluation. The most simple decision logic is to compare the evaluated residual signal with the threshold. If the evaluated residual exceeds the threshold then the fault-alarm is released. For a specific fault, it is possible to determine its location using the following thresholds:

$$\begin{aligned} |r_k(t)| > \ T_k &\Rightarrow f_k(t) = 1 \\ |r_k(t)| \leq \ T_k &\Rightarrow f_k(t) = 0 \end{aligned} \tag{4.44}$$

where the threshold $T_k = 1\text{x}10^5$, $f_k(t)$ is the sensor fault and k =1, 2, 3, 4 (for pitch angle, rotor speed, generator speed and torque sensors, respectively).

The selection of the threshold is very important. If the chosen threshold is too low, it results in false alarms, i.e. some disturbances will cause the residual to cross the

threshold and result in an alarm. If the selected threshold is too high, small faults are undetected. Usually, in deterministic settings, the selected threshold is slightly higher than the value of the evaluated residual signal in a fault-free case.

4.5 Conclusions

An observer-based residual generator was designed using a set of structured residual generators for the assessment of the performance of this method in wind turbines. Simulation results demonstrated that the method is suitable for the detection of faults in sensors actuators and components. However, it did not achieve fault isolation.

The contribution of this chapter was to develop a robust observer-based sensor fault detection and isolation scheme for wind turbines to accomplish fault detection and isolation. This scheme is systematic and easy to design and implement. Simulation results demonstrated that it is suitable for the detection and isolation faults in sensors and, also, that it is simple to handle multiple faults. The advantage of the proposed approach is that it depends on only one observer compared with other approaches, using a bank of observers such as a structured residual set designed by a dedicated or a generalised observer scheme.

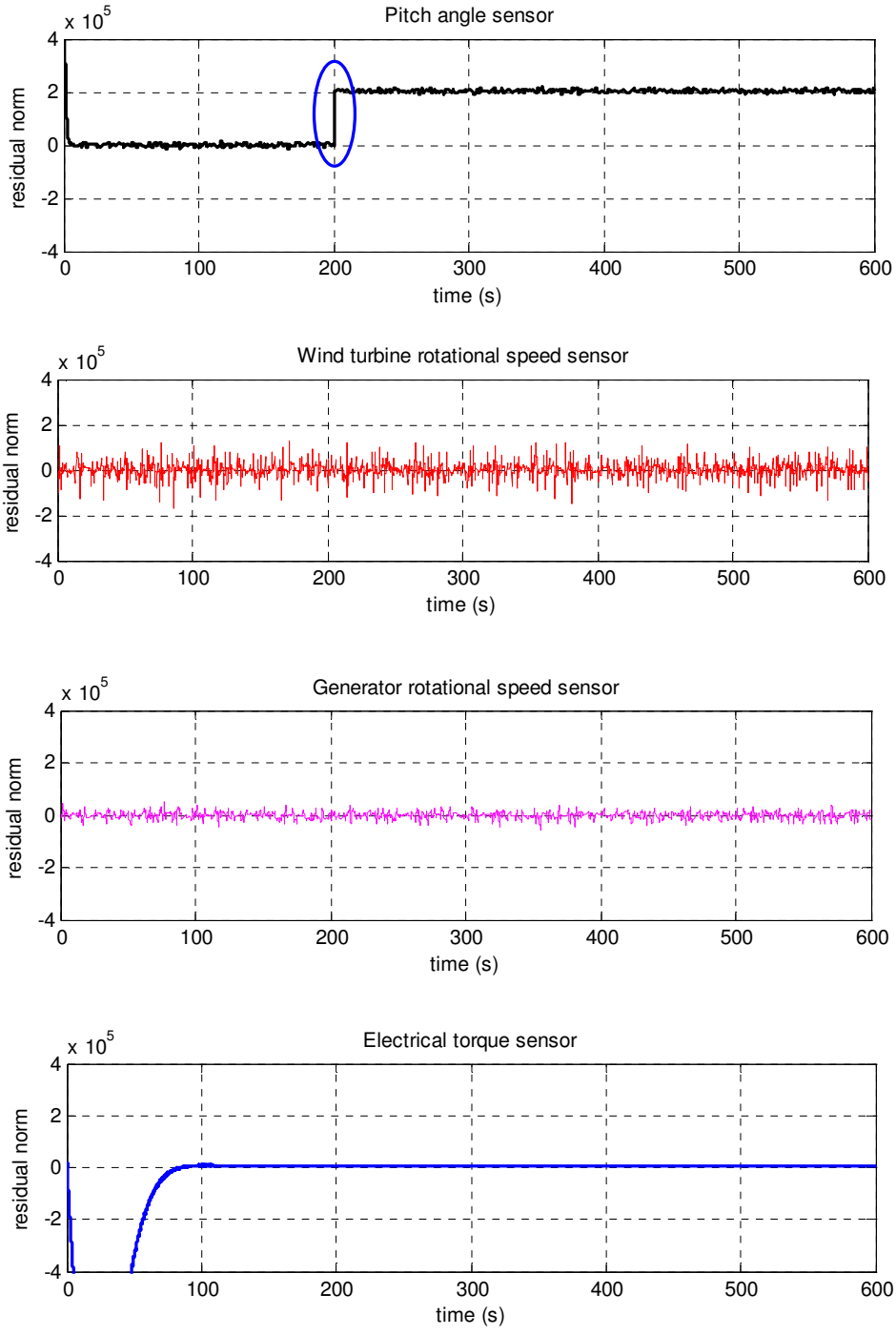


Figure 4.19: The residual norm, when a fault occurs in the pitch angle sensor

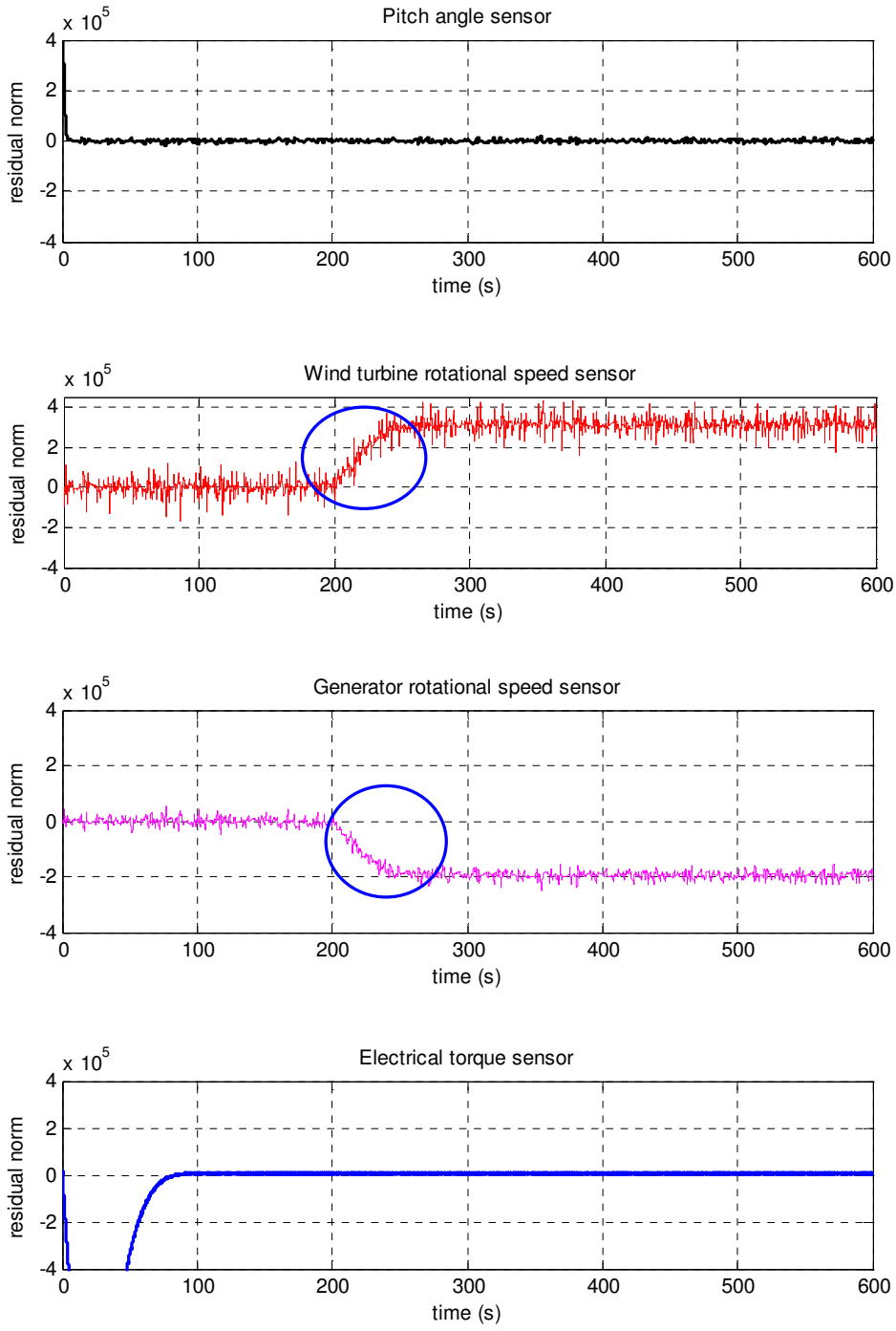


Figure 4.20: The residual norm, when a fault occurs in the rotor speed sensor

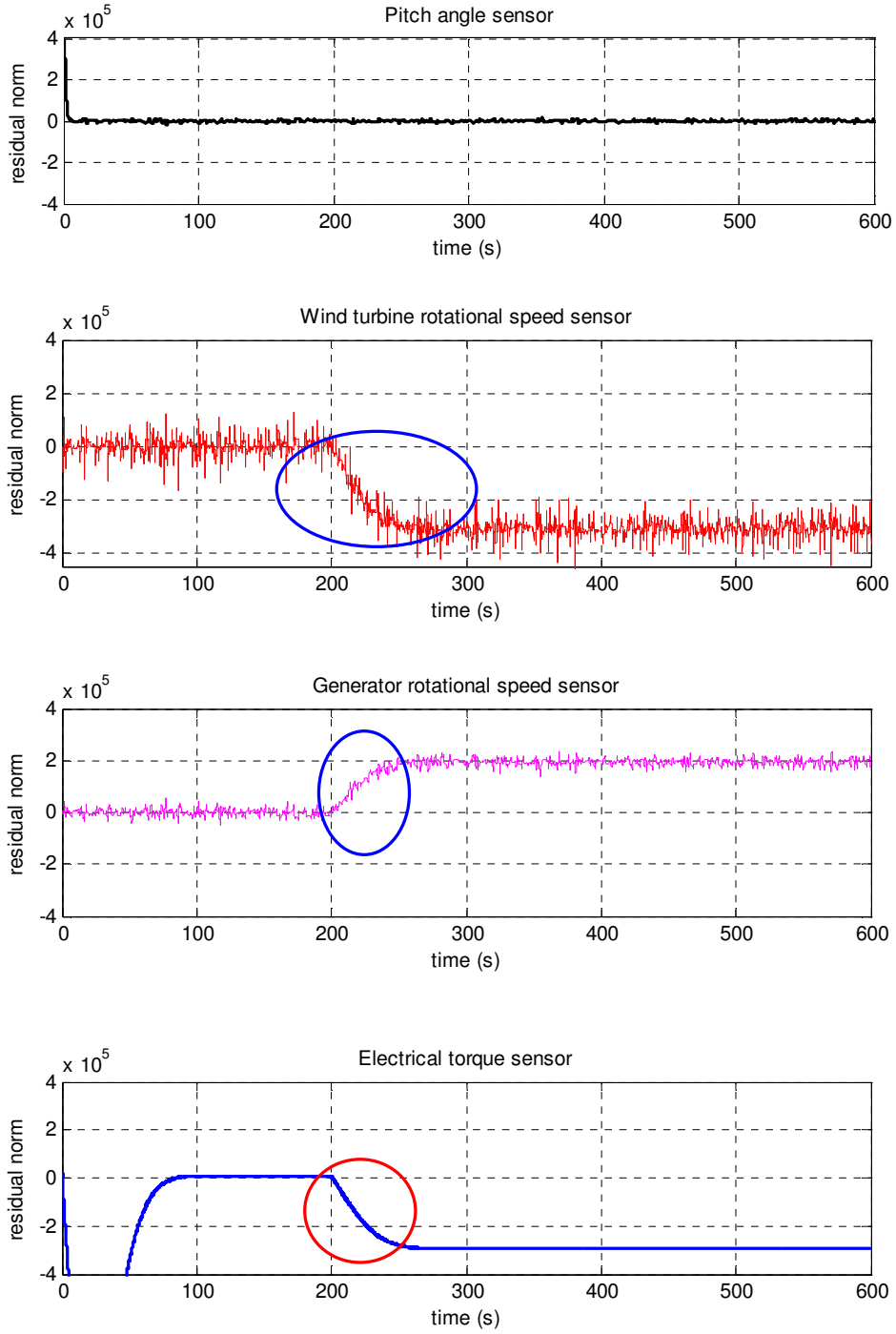


Figure 4.21: The residual norm, when a fault occurs in the generator speed sensor

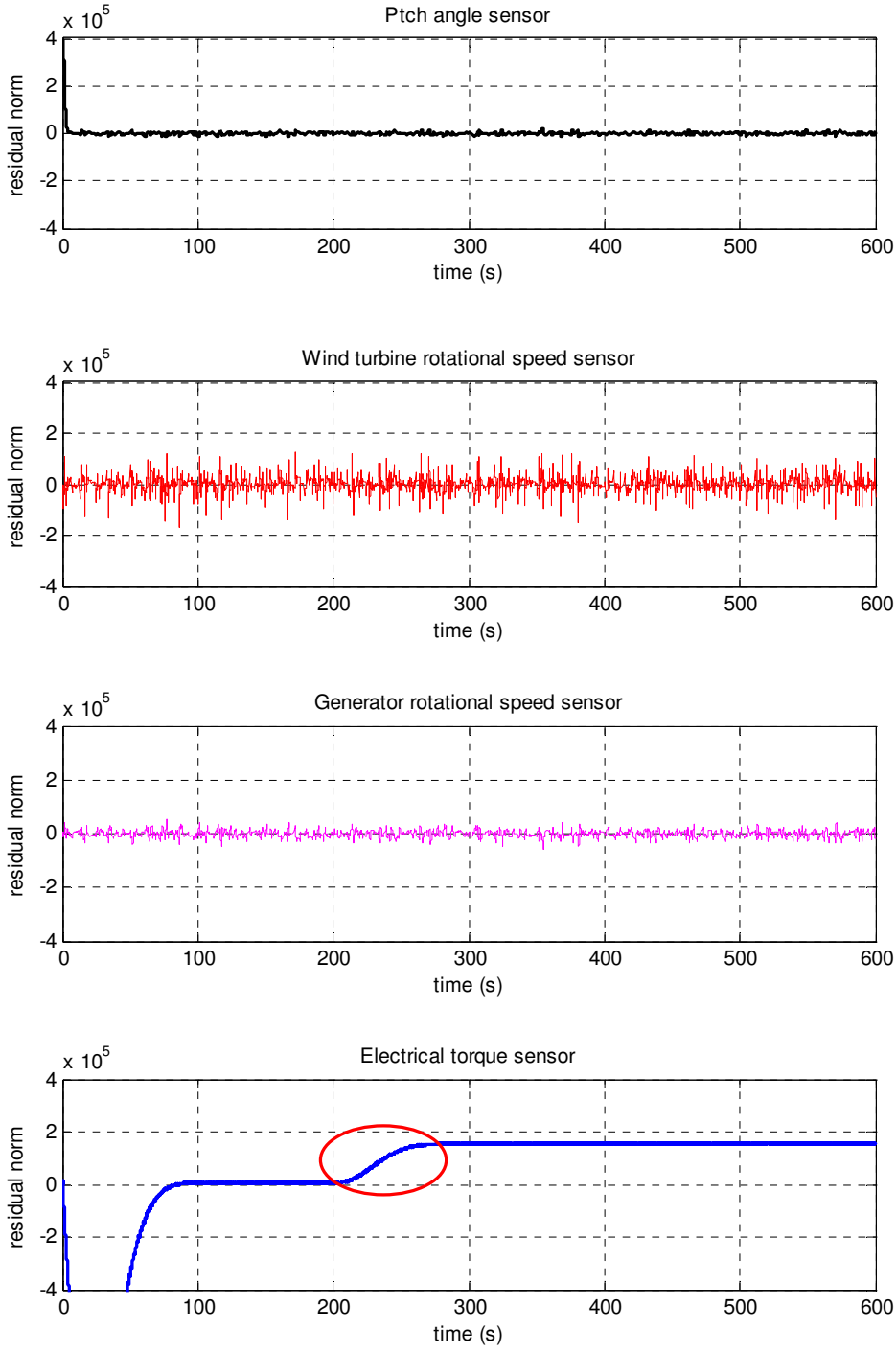


Figure 4.22: The residual norm, when a fault occurs in the generator torque sensor

5 Non-linear Observer-based Fault Detection and Isolation

5.1 Introduction

Linear systems have been studied extensively and have proven to be extremely useful. Applying observer-based fault detection to the linear system has been successful, as illustrated in the previous chapter. If the process is strongly non-linear or if the operating region is too broad, the fault detection techniques must handle a large linearisation error. Therefore, there is a necessity to study fault detection techniques for non-linear systems. Fault detection in dynamical systems continues to be an active field of research and there is a clear need for improvements.

This chapter is organised into three topics. The first section describes how a non-linear observer wind turbine model and residual equation are formed. The second section explains the non-linear observer methodologies that can be applied to fault detection. The third section explains the contribution of this thesis to designing a state-dependent non-linear observer for detecting and isolating faults in wind turbines.

5.2 Preface of a non-linear observer wind turbine model

In the previous chapter (linear state-space wind turbine model), the state of the wind turbine linear model was defined by the following equation:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

An observer was constructed that has the same structure as the linear model with the addition of a driving feedback term whose role is to reduce the observation error to zero, as given by:

$$\hat{x}(t) = A\hat{x}(t) + Bu(t) + K(y(t) - \hat{y}(t))$$
$$y(t) = C\hat{x}(t)$$

Theoretically, studying observers for non-linear systems is much more difficult than studying observers for linear systems; however, the same logic can be used to construct a non-linear observer. The non-linear wind turbine model can be represented as:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned} \tag{5.1}$$

$x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, $f(.)$ and $g(.)$ are non-linear vector functions of dimensions *n* and *p*, respectively. Based on the knowledge of linear observers, the following structure for a non-linear observer can be proposed:

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x},u) + K(\hat{x},u)(y - \hat{y}) \\ \hat{y} &= g(\hat{x},u) \end{aligned} \tag{5.2}$$

The observation error *e(t)* tends to zero, at least at the steady state:

$$e(t) = x(t) - \hat{x}(t) \tag{5.3}$$

The observation error dynamics can be determined by:

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) = f(x,u) - f(\hat{x},u) - K(\hat{x},u)(g(x,u) - g(\hat{x},u)) \tag{5.4}$$

By eliminating $\hat{x}(t)$ from the error equation, the following is obtained:

$$\dot{e} = f(x,u) - f(x-e,u) - K(\hat{x},u)(g(x,u) - g(x-e,u)) \tag{5.5}$$

It is obvious that $e(t) = 0$ is the solution of differential equation; this indicates that the constructed observer may have $e = 0$ at the steady state. The gain must be chosen such that the observer and error dynamics are stable asymptotically in order to force the error at the steady state to $e = 0$. The asymptotic stability can be examined using Lyapunov's first method for stability.

5.3 Residual generation

Generation of residuals (signals which contain information about the failures or defects) reflects the faults and is achieved by estimating the outputs of the process and using the estimation errors as the residuals. For the fault detection task, a single observer or Kalman filter is sufficient. For fault isolation, correctly structured sets of residuals are required; one may use linear or non-linear, full or reduced-order, fixed or adaptive observers, or Kalman filters.

A number of methods have been proposed in recent decades for an observer-based residual generation. The most significant approaches are the fault detection filter, the dedicated and generalised observer scheme, and the unknown input observer scheme. The design of robust residual generators which are invariant or at least insensitive with respect to unknown inputs has been an area of increasing focus.

The basic idea of residuals is derived from implicit information in functional relationships; this exists between measurements taken from the process and a process model. In this sense, a residual is a fault indicator based on the difference between measurements and model-based computations. Model-based diagnoses use models to obtain residual signals which are, as a rule, zero in the fault-free case and non-zero otherwise. Faults are usually detected by setting a threshold on a residual signal generated from the difference between real measurements and their estimates using the mathematical model.

Considering processes or sensor faults; the non-linear model given in equation (5.1) can be modified as follows:

$$\begin{aligned} \dot{x} &= f(x,u) + R_1\ f(t) \\ y &= g(x,u) + R_2 f(t) \end{aligned} \tag{5.6}$$

Matrices R_1 and R_2 are the fault distribution matrices. Based on equations (5.2) and (5.6), the state estimation error is given by:

$$\dot{e} = f(x,u) - f(x-e,u) - K(\hat{x},u)(g(x,u) - g(x-e,u)) - K(\hat{x},u)R_2 f(t) + R_1 f(t) \tag{5.7}$$

The residual is the output estimation error:

$$r = y - \hat{y} = g(x,u) - g(x-e,u) + R_2 f(t) \tag{5.8}$$

Residual *r* depends directly upon $R_1(t) \in R^n$ (actuator or component failures) and $R_2(t) \in R^p$ (sensor failures). *f(t)* represents a fault vector.

5.4 Methodologies of non-linear observer

Several observer-based approaches have been proposed in recent years for residual generation in non-linear systems. These approaches, which are applied particularly to fault detection, are now described.

5.4.1 Lyapunov-based method

Adjallah et al. (1994) [102] proposed the so-called Lyapunov-based method for the detection and isolation of sensor faults. They designed a non-linear observer in order to achieve fault detection and localisation for a wide class of non-linear systems subjected to bounded non-linearities. They proposed a dedicated non-linear observer scheme for fault detection and identification. The basic idea of this approach is to reconstruct the state and output of the process under consideration and then to analyse the output estimation error.

The gain matrix $K(\hat{x},u)$ in equation (5.2) is determined using the Lyapunov function in equation (5.9), as shown by the following steps:

$$\dot{V}(e) = e^T P \dot{e} \tag{5.9}$$

where the matrix $P=P^T > 0$ is symmetric positive and definite. The dynamics of the estimation error can be rewritten as in the following equation (refer to [102]):

$$\dot{e} = [\frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} - K(\hat{x},u)C]e \tag{5.10}$$

In [103], an algorithm is proposed to determine the gain $K(\hat{x},u)$ based on the assumption that $K_{error}C \neq 0$. The algorithm consists of two steps: (i) determination of *P* and (ii) determination of $K(\hat{x},u)$ using the previous value of *P*.

1. *Step 1:*

If $e \in K_{error}(C) = 0$, then, by substituting equation (5.10) in equation (5.9), equation (5.9) is reduced to:

$$\dot{V}(e) = e^T P \frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} e \tag{5.11}$$

Next, a matrix P is found, which ensures the condition:

$$\dot{V}(e) = e^T P \frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} e < 0 \tag{5.12}$$

Solving equation (5.12) yields a value for P.

2. *Step 2:*

$K(\hat{x},u)$ is determined, verifying the following inequality:

$$\dot{V}(e) = e^T P[\frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} - K(\hat{x},u)C]e < 0 \tag{5.13}$$

A sufficient condition to reach this inequality is:

$$\frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} - K(\hat{x},u)C < 0 \tag{5.14}$$

This is achieved by using first the following structure for $K(\hat{x},u)$:

$$K(\hat{x},u) = P^{-1}F(\hat{x},u)C^T Q \tag{5.15}$$

where $F(\hat{x},u)$ and Q are n- and p-dimensional square matrices to be determined, respectively. Substituting equation (5.15) in equation (5.13) gives:

$$\dot{V}(e) = e^T P \frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} e - e^T F(\hat{x},u) C^T Q C e < 0 \qquad (5.16)$$

A map $F(\hat{x},u)$ which satisfies the inequality (5.16) is positive definite and is defined for all t such that:

$$| e^T P \frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} e | < e^T F(\hat{x},u) C^T Q C e < 0 \qquad (5.17)$$

Secondly, assuming that such a map exists, then equation (5.16) can be rewritten as:

$$\dot{V}(e) \leq | e^T P \frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} e | - e^T F(\hat{x},u) \frac{\partial g^T}{\partial x^T}\Big|_{x=\hat{x}} Q \frac{\partial g}{\partial x^T}\Big|_{x=\hat{x}} e < 0 \qquad (5.18)$$

According to inequality (5.17), a sufficient condition for fulfilling the Lyapunov stability condition (5.12) can be summarised as follows:

find a (*pxp*) matrix Q satisfying $C^T Q C - I \geq 0$.

All positively defined matrices *F(x, u)* which verify the inequality:

$$\| P \frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}} \| < \| F(\hat{x},u) \| \qquad (5.19)$$

satisfy inequality (5.17). *F(x, u)* can be calculated as the following map [102]:

$$F(x,u) = diag \left\{ \frac{1}{2} \sum_{j=1}^{n} \left| \alpha_{ij} + \alpha_{ji} \right| \right\} \qquad (5.20)$$

where α_{ij} are the elements of the matrix $P \frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}}$

By satisfying the above conditions, the state observer can be described by:

$$\dot{\hat{x}} = f(\hat{x},u) + P^{-1} F(\hat{x},u) C^T Q (y - C\hat{x}) \qquad (5.21)$$

With regard to the existence of process or sensor faults, as in equation (5.6), the differential equation governing the dynamics of the state estimation error is:

$$\dot{e}=[\frac{\partial f(x,u)}{\partial x^T}\Big|_{x=\hat{x}}-K(\hat{x},u)C]e-K(\hat{x},u)R_2 f(t)+R_1 f(t) \qquad (5.22)$$

It is clear from equation (5.22) that the output estimation error is a function of R_1 and R_2. Consequently, the faults affect the output estimation error. Therefore, the latter can be used as a residual to indicate that a fault has occurred.

This approach has been applied to a non-linear model of a synchronous machine [102]; experimental results were used to illustrate the application of a dedicated non-linear observer scheme to isolate sensor faults. In contrast to linearised systems, the resulting non-linear observer is robust with respect to the operating point of the non-linear system.

5.4.2 Tsinias observer for non-linear systems

Tsinias (1989) [103] proposed, as a direct extension to the observer in the linear case, a dynamic observer that was suitable for application to a wide class of non-linear systems subjected to bounded non-linearities. This approach provided sufficient conditions and an extremely simple approach for the observer design. The main sufficient condition, given below, is that of the Lyapunov type and it turns out, also, to be necessary in the linear case. In addition, it should be pointed out that Lyapunov conditions were used successfully to determine stabilising feedback laws for non-linear systems. The trajectories *e(t)* of the error equation are defined for all positive *t* approaching zero as $t \to \infty$ for any initial *e(0)* and the observer performs globally the state determination of the system. Tsinias (1989) considered non-linear systems of the form:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= Cx \end{aligned} \qquad (5.23)$$

It is assumed that for any input *u* and initial state x_0, the corresponding trajectory *x (t, x_0, u)* of equation (5.23) is defined for all positive *t*. Finally, mapping (*x, u*) is continuously differentiable. Tsinias considered an observer in the following form:

$$\dot{\hat{x}} = f(\hat{x},u) + K(u)(y - C\hat{x}) \qquad (5.24)$$

Then, the dynamic error is:

$$\dot{e} = (f(x,u) - f(\hat{x},u)) - K(u)Ce \tag{5.25}$$

The state determination of equation (5.23) is possible if the following conditions are fulfilled.

Condition 1:

There exists a positive definite matrix P of dimension $n \times n$ and a positive constant k_1 such that for any $e \in K_{error}\ C=\{0\}$, the following is satisfied:

$$e^T P\{D_x f(x,u)\}e \le -k_1 \|e\|^2 \tag{5.26}$$

where ||.|| denotes the norm.

Condition 2:

There exists a continuous real function p and a positive constant k_2 such that $p(u) \geq k_2$, and:

$$|e^T P\{D_x f(x,u)\}e| \le p(u)\|e\|^2 \tag{5.27}$$

Suppose that Conditions 1 and 2 are satisfied, then:

For any sufficiently large positive constant h, the error equation (5.25) holds with:

$$K(u) = hp(u)P^{-1}C^T \tag{5.28}$$

where p and P are given in Conditions 1 and 2, and perform the state determination of equation (5.23).

To define h, Tsinias (1989) [103] considered an open sphere S_p of radius $r > 0$, centred at zero. Let $h = \partial S_p \cap K_{error}C$, ∂S_p be the boundary of S_p. Then, h is compact and, by use of the Condition 1, there exists an open set M such that $h \subset \partial S_p \cap M$ and, the Lyapunov function is used for each (x, u) provided that $h > L_1 L_2^{-1}$

where:

$$L_1 = \sup_{\substack{e \in \partial s_p - (\partial s_p \cap M) \\ (x,u) \in R^n x U}} \frac{e^T P\{D_x f(x,u)\}e}{p(u)} + \frac{k_1 \|e\|^2}{p(u)} \le r^2\left(1 + \frac{K_1}{K_2}\right) < \infty \tag{5.29}$$

and:

$$L_2 = \min_{e \in \partial s_p - (\partial s_p \cap M)} \|Ce\|^2 > 0 \tag{5.30}$$

Consider that the system is observable and satisfies the above two conditions. There is a positive constant *h* such that the linear system can be described by equation (5.31), which is an observer for (5.23):

$$\dot{\hat{x}} = A\hat{x} + Bu + h\, P^{-1} C^T (y - C\hat{x}) \tag{5.31}$$

When compared to the previous mentioned approach (Lyapunov-based method), the main result presents some advantages because it is a direct extension of the observer design in the linear case; it is much simpler to follow and the observer's dynamics are at least continuous mappings at the origin, where the error equation is stable both globally and asymptotically. However, the approach requires exact knowledge of the plant's non-linearities.

It can be concluded that, if the conditions are satisfied, this approach can be developed to apply to a wide class of non-linear systems. Direct extension of a dynamic observer in linear cases can be used to detect actuator sensor component faults, as in the case of a linear observer-based FDI scheme.

5.4.3 Thau non-linear observer approach

Thau, in 1973 [104], proposed a different approach to observing the states of a non-linear system. This method did not include a systematic technique for the construction of the observer. Instead, it gave a sufficient condition for asymptotic stability of the origin of the error differential equation. Using this method, the error between the system and the output of the observer is shown to be convergent asymptotically to zero, provided that an additional assumption is valid. The class of non-linear systems examined in this approach is described by:

$$\begin{aligned} \dot{x} &= Ax + f(x) + Bu \\ y &= Cx \end{aligned} \tag{5.32}$$

The non-linear function *f(x)* may contain, also, linear terms in *x*. Assuming (A, C) is observable completely, *K* can be found such that the eigenvalues of $A_o = A - KC$ are in the left half plane (LHP).

The estimate of the true state is shown by the following equation:

$$\begin{aligned}\dot{\hat{x}} &= A_o\hat{x} + f(\hat{x}) + Bu + Ky \\ y &= C\hat{x}\end{aligned} \tag{5.33}$$

Let error *(e)* be given by:

$$e = \hat{x} - x \tag{5.34}$$

Thus, *e* satisfies the differential equation:

$$\dot{e} = A_o e + f(\hat{x}) - f(x) = A_o e + f(x+e) - f(x) \tag{5.35}$$

Since the eigenvalues of A_o are in the LHP, for any given positive definite $Q \in \mathbb{R}^{nxn}$ there exists a unique positive definite $P \in \mathbb{R}^{nxn}$ which satisfies the equation (5.36) [106]:

$$A_o^T P + PA_o = -2Q \tag{5.36}$$

Next, consider the following positive definite Lyapunov function candidate:

$$V(e) = e^T Pe \tag{5.37}$$

The derivative of *V(e)*, evaluated along the solution of the error differential equation (5.35) is given by:

$$\dot{V}(e) = \dot{e}^T Pe + e^T P\dot{e} = -2e^T Qe + 2e^T p[f(x+e) - f(x)] \tag{5.38}$$

Assume the additional constraint that the function *f(x)* is Lipschitz locally about the origin; that is, there exists a positive constant *L* such that:

$$\|f(x_1) - f(x_2)\| \le L\|x_1 - x_2\| \tag{5.39}$$

For all x_1, x_2 in some open region (R_o) contains the origin. Therefore, if *e* is contained in R_o, then, the following inequalities are valid:

$$\dot{V}(e) \le -2e^T Qe + 2L\|Pe\| \ \|e\| \le (-2a + 2L\|P\|\ )\|e\| \tag{5.40}$$

where *a* is the minimum eigenvalue of *Q* and $\|P\|$ is the maximum eigenvalue of *P*. Hence, if:

$$a > \|P\| > L \tag{5.41}$$

Then $e = 0$ is an asymptotically stable equilibrium point of equation (5.35).

The Thau observer guarantees that the observer error is convergent globally and asymptotically to zero, as illustrated above. However, it may not be possible to satisfy the sufficient condition given in equation (5.41). Also, the Thau observer includes the non-linearity of the system into the observer design, and, therefore, it is essential to have certain knowledge of this non-linearity.

Based on Thau observers, the principle for designing a non-linear observer for residual generator, designed for one class of continuous-time non-linear multiple input and multiple output systems, is shown in [126]. The essential aim was to design a fault-tolerant control which was achieved by the conception of a diagnosis procedure. This procedure generated a residual signal which was then evaluated within decision functions.

5.4.4 Extended Luenberger observer-based method

In order to determine $K(\hat{x},u)$, the observer equation (5.2) is transformed into the canonical coordinates ($\hat{x}^*(\hat{x}=T(\hat{x}^*,\bar{u}))$, $T \in \mathbb{R}^{nxn}$ is a non-linear transformation. Then, through an extended linearisation about the reconstructed trajectory $\hat{x}^*$ of this equation, the observer gain is defined according to Zeitz (1987) [107], as in equation (5.42):

$$K(\hat{x},u) = \frac{[p_1\ell_f^0 + p_2\ell_f + p_3\ell_f^2 + p_n\ell_f^{n-1} + \ell_f^n]\left(\overline{\frac{\partial g^*}{\partial \hat{x}_n^*}} Q^{-1}(\hat{x},\bar{u})\begin{bmatrix}0 & \cdots & 0 & 1\end{bmatrix}^T\right)}{\overline{\frac{\partial g^*}{\partial \hat{x}_n^*}}} \tag{5.42}$$

where $x_n^* = g^{*-1}(y,u)$ and $y = g^*(x_n^*,u)$ is introduced for the equation (5.1), $Q^{-1}(\hat{x},\bar{u})$ is the inverse of the observability matrix $\bar{u} = [u,\dot{u},\ldots,u^{(n-1)}]^T$ and $p = [p_1, p_2, \cdots, p_n]^T$ is the coefficient of the polynomial characteristic. The linear differential operator ℓ_f is defined according to equation (5.43) [107]:

$$
\begin{aligned}
&\ell_f^0 \frac{\overline{\partial T}}{\partial x_1^*} = \frac{\overline{\partial T}}{\partial x_1^*} \\
&\ell_f^i \frac{\overline{\partial T}}{\partial x_1^*} = \frac{\overline{\partial T}}{\partial x_{i+1}^*} \qquad i = 1, \cdots, n-1 \\
&\ell_f^n \frac{\overline{\partial T}}{\partial x_1^*} = \frac{\overline{\partial T}}{\partial x^*} \frac{\overline{\partial f}}{\partial y} \frac{\overline{\partial g}}{\partial x_n^*}
\end{aligned}
\tag{5.43}
$$

$K(\hat{x}, u)$ is computed at each time instant; this achieves stability since the requirements of repetitive calculations of observer gain result in more online computations and linearisation errors.

An extended Luenberger observer approach was applied to detect incipient failures automatically in the feedback sensors of the control systems [127]. The feasibility of the scheme was investigated by applying it to a simplified version (fourth order) of the control system. A single set of inertial instruments was used to provide the feedback signals; the redundancy, which is obtained normally by multiple instrument sets, was obtained here, artificially, by a subsystem of multiple Luenberger observers and logic circuits. Tests indicated that scale factor errors, errors due to threshold effects and bias errors in the instruments were detected as they occurred.

5.4.5 State-dependent non-linear observer

The state-dependent Riccati equation (SDRE) method was demonstrated empirically in a number of applications. The method is based on extended linearisation of the process dynamics [108, 109]. The dynamics are expressed by:

$$
\begin{aligned}
\dot{x} &= A(x)x + B(x)u \\
y &= C(x)x
\end{aligned}
\tag{5.44}
$$

The matrices *A(x)*, *B(x)* and *C(x)* are not unique; efficient selection of these matrices is known as the parameterisation problem [110] and may affect the performance of the observer.

For a non-linear observer, the estimated state $\hat{x}$ is given by:

$$\dot{\hat{x}} = A(\hat{x})\hat{x} + K(\hat{x})[y - C(\hat{x})\hat{x}] \tag{5.45}$$

The non-linear observer gain matrix $K(\hat{x})$ can be computed using the state-dependent algebraic Riccati equation (SDARE):

$$K(\hat{x}) = P(\hat{x})C^T(\hat{x})R^{-1} \tag{5.46}$$

with:

$$P(\hat{x})A^T(\hat{x}) + A(\hat{x})P(\hat{x}) - P(\hat{x})C^T(\hat{x})R^{-1}C(\hat{x})P(\hat{x}) + Q = 0 \tag{5.47}$$

Q(x) and *R(x)* are design matrices selected, as in the case of a linear system, to give weight to the state and the observer, respectively.

In some applications, the use of the state-dependent algebraic Riccati equation can impose an overly restrictive requirement on the observability and controllability of the non-linear system. Therefore, Haessig et al. [111] suggested the state-dependent differential Riccati equation (SDDRE) rather than the algebraic Riccati equation (the state estimates do converge to the actual states more rapidly, compared with the SDARE observers), which is presented in equation (5.48). The solution of SDDRE can be found by using numerical integration methods:

$$\dot{P}(\hat{x}) = P(\hat{x})A^T(\hat{x}) + A(\hat{x})P(\hat{x}) - P(\hat{x})C^T(\hat{x})R^{-1}C(\hat{x})P(\hat{x}) + Q \tag{5.48}$$

By exploring SDDRE techniques for non-linear observer designs, it was noted that all previous applications of SDRE concerned control design. In further work, this approach is applied here to fault detection for a non-linear wind turbine.

5.4.6 Construct the non-linearity from direct measurements-based method

Heraud et al. (2010) [105] proposed a non-linear observer-based approach to the fault detection and isolation issue of wind turbine systems. They used a DFIG and a non-linear observer was employed to generate the residual for fault detection purposes. Tests were performed using the experimental benchmark. It consisted of a DC machine of 25 kW, emulating the aerodynamic and mechanical behaviour of a wind turbine, and a 15 kW DFIG,

emulating the electric generator. This benchmark presented physical characteristics that were close to those of most existing wind generators.

Bennouna et al. (2009) [112] used the DFIG model to assimilate the speed of the constant wind and, then, they supposed that the generator's rotor speed was steady; that is why the wind turbine linear model was defined as in equation (5.7). However, in the real case, the generator's rotor speed (ω_m) is variable. Therefore, Heraud et al. (2010) [105] considered that equation (5.7) could be rewritten in the non-linear state form as:

$$\begin{aligned} \dot{x} &= A(\omega_m)x + Bu \\ y &= Cx \end{aligned} \tag{5.49}$$

This representation of the system is non-linear because *A* is not a constant matrix since it depends on ω_m. The system seems to belong to a class of systems in which the observability does not depend on the input; systems of this class are known as uniformly observable systems. They allow a gain to be designed, as in Luenberger observers, which either compensate all non-linear elements exactly by output injection or dominate them via the linear part. Here, the non-linearity can be constructed from direct measurements and, thus, compensated in the observer design as proposed in [113, 114, 115]. Then, equation (5.49) can be changed as below:

$$\begin{aligned} \dot{x} &= \bar{A}x + \varphi(y,u) \\ y &= Cx \end{aligned} \tag{5.50}$$

If $(\bar{A}, C)$ is observable, the system (5.50) admits an observer of the form:

$$\begin{aligned} \dot{\hat{x}} &= \bar{A}\hat{x} + \varphi(y,u) - K(\hat{y} - y) \\ \hat{y} &= C\hat{x} \end{aligned} \tag{5.51}$$

with *K* such that $(\bar{A} - KC)$ is stable and φ depends only on *u* and *y*, which are directly available.

The overseen conditions of observability and stability, equations (5.50) and (5.51), allow us to write the observer in the form:

$$\dot{\hat{x}} = (\bar{A} - KC)\hat{x} + \varphi(y,u) + Ky \tag{5.52}$$

The error of state estimation is given by:

$$e = \hat{x} - x \tag{5.53}$$

Then, using equations (5.50) and (5.51), we obtain:

$$\dot{e} = (\bar{A} - KC)(\hat{x} - x) = (\bar{A} - KC)e \tag{5.54}$$

The residue is chosen as:

$$r = \hat{y} - y = Ce \tag{5.55}$$

Note that *r* possesses the characteristic features of a residual when the observer matrix *K* is chosen so that $(\bar{A} - KC)$ is stable. In this case, $\hat{x}$ provides, also, an unbiased estimation for *x*.

This method presented a non-linear observer to solve fault detection and isolation of a wind generator, especially the DFIG, which is the most commonly used modern generator. The system is totally observable. This method allows one to estimate a state using a non-linear observer. It is concluded that the DFIG model and the observer match well to the wind turbine model. In further work, this approach can be applied to monitoring other wind turbine components such as the drive-train and the wind turbine rotor.

5.4.7 Extended Kalman filter-based method

The extended Kalman filter (EKF) has become a standard technique for parameter and state estimation in non-linear systems [117, 118]. The EKF uses a basic process model to make an estimate of the current state of a system and, then, corrects the estimate using any available sensor measurements. The basic role of the filter is to find the states and parameters of a non-linear dynamic system. Discrete-time non-linear dynamic systems can be described by a dynamic state-space model of the following form:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k) \\ z_k &= h(x_k, v_k) \end{aligned} \tag{5.56}$$

where x_k is the state of the system including any unknown model parameters, y_k is the measurement of the system, u_k is the input of the system and w_k and v_k represent the process noise and observation noise, respectively.

To design an EKF equation, (5.56) is linearised about the operating points, $\hat{x}_{k-1|_{k-1}}$ and $\hat{x}_{k|_{k-1}}$, respectively:

$$\begin{aligned} x_k &= \hat{x}_{k_{|k-1}} + A(x_{k-1} - \hat{x}_{k-1_{|k-1}}) + Ww_{k-1} \\ z_k &= h(\hat{x}_{k_{|k-1}}) + H(x_k - \hat{x}_{k_{|k-1}}) + Vv_k \end{aligned} \tag{5.57}$$

where:

$$A_{i,j} = \frac{df_i}{dx_j}(\hat{x}_{k-1_{|k-1}}, u_k) \tag{5.58}$$

$$W_{i,j} = \frac{df_i}{dw_j}(\hat{x}_{k-1_{|k-1}}, u_k) \tag{5.59}$$

$$H_{i,j} = \frac{df_i}{dx_j}(\hat{x}_{k_{|k-1}}) \tag{5.60}$$

$$V_{i,j} = \frac{dh_i}{dv_j}(\hat{x}_{k_{|k-1}}) \tag{5.61}$$

The EKF's basic operation involves two steps: prediction and correction. In the prediction step, the state and covariance estimates from the previous time step, $\hat{x}_{k-1_{|k-1}}$ and $P_{k-1_{|k-1}}$, are projected to the current time step:

$$\hat{x}_{k_{|k-1}} = f(\hat{x}_{k-1_{|k-1}}, u_k) \tag{5.62}$$

$$P_{k_{|k-1}} = A_k P_{k-1_{|k-1}} A_k^T + W_k Q_{k-1} W_k^T \tag{5.63}$$

In the correction step, the state and the covariance matrices are corrected with the measurements:

$$K_k = P_{k_{|k-1}} H_k^T (H_k P_{k_{|k-1}} + V_k R_k V_k^T)^{-1} \tag{5.64}$$

$$\hat{x}_{k_{|k}} = \hat{x}_{k_{|k-1}} + K_k(z_k - h(\hat{x}_{k_{|k-1}})) \tag{5.65}$$

$$P_{k_{|k}} = (I - K_k H_k) P_{k_{|k-1}} \tag{5.66}$$

where R_k and Q_k are the measurement and process noise covariance matrices, respectively.

The predicted output based on the EKF state estimate is given by:

$$\hat{z}_k = h(\hat{x}_{k|_{k-1}}) \tag{5.67}$$

Then, the residual is:

$$r_k = z_k - \hat{z}_k \tag{5.68}$$

The EKF relies on first-order approximations of the non-linear system. These approximations can cause substantive errors, which may lead to poor filter performance.

The standard approach of using the EKF filter, as a FDI tool, is to design a bank of Kalman filters where, as shown in Figure 5.1, each filter is tuned to a specific fault. The normal EKF model provides the best estimate for the system under nominal operating conditions. Other EKF models integrate the fault effects in their structures and represent different fault types. Each of these filters is built with an embedded fault model and an increase in robustness is expected since each filter uses additional knowledge about the expected fault. Since each Kalman filter is sensitive to a particular fault, a small residual would suggest a match between the actual measurement and the corresponding filter and, therefore, a possible fault. Consequently, the size of the residuals from various filters gives a relative indication of how each of these models represents the actual fault status sufficiently accurately. Therefore, this multiple model approach achieves, also, fault isolation.

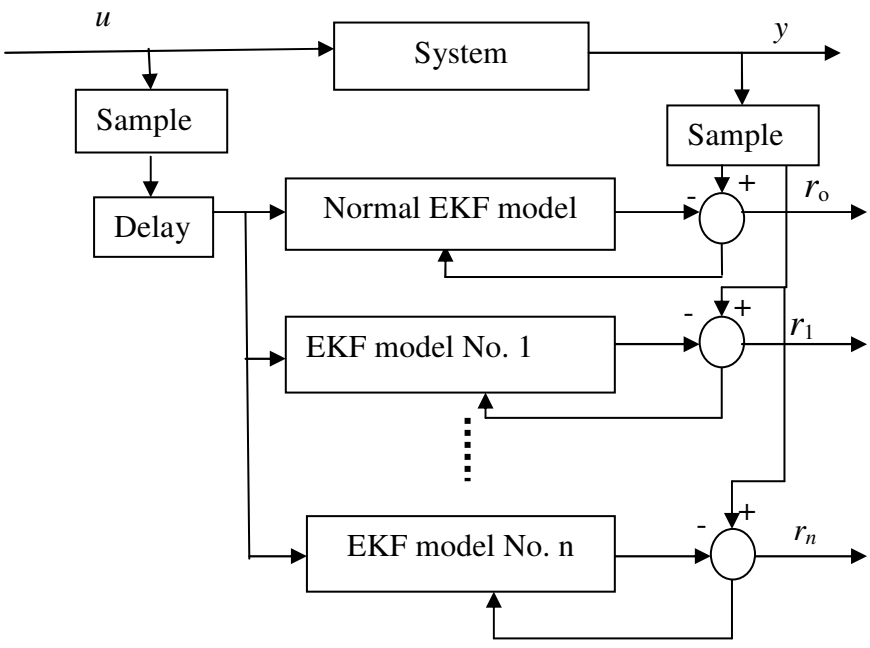


Figure 5.1: Multiple model Kalman filter approach to FDI system structure [120]

Azad et. al. (2011) [116] investigated the EKF Kalman filter-based estimation technique for parameter estimation of the DFIG driven by the wind turbine. The DFIG parameters were

influenced by various factors including temperature, magnetic saturation and eddy currents. Here, the EKF technique was evaluated from three different aspects: (i) estimation accuracy and computation time, (ii) robustness to changes of the initial estimates of the parameters and (iii) robustness to changes of the filter gains. They also studied the robustness of the algorithm to variations of the initial parameter estimates and noise covariance matrices Q and R.

The EKF aims to approximate the non-Gaussian density with the Gaussian one; this may cause several problems such as an unbounded error variance growth [120]. The reason is that this filter is based on the Kalman filter, which is an optimal estimator only if the noise is Gaussian.

5.4.8 Unscented Kalman filter-based method

The unscented Kalman filter (UKF) is a non-linear filter, first proposed by Julier and Uhlmann [121], to deal with the problems which can appear in EKF estimation due to linearisation. The UKF was proposed to be a better alternative to the EKF in state estimation for a variety of application fields. The UKF is founded on the concept that it is easier to approximate a Gaussian distribution than it is to approximate arbitrary non-linear functions. The key to UKF is a sampling technique, known as an unscented transform, which chooses a set of sample points to estimate the probabilistic model of the state evolution. These sample points are known as 'sigma points' and they capture the true mean and covariance of the posterior probability density function of the states. These points are propagated through the full non-linear system model from which the posterior mean and covariance are obtained. The basic operation of the UKF involves two steps as in EKF: prediction and correction. In the prediction step, a number of sigma points are selected and, then, the state and covariance matrices are estimated as the weighted mean of these sigma points. To find the sigma points first, the estimated state and covariance matrices are augmented with the mean and covariance of the process noise:

$$x^{a}_{k-1|k-1} = \left[\hat{x}^{T}_{k-1|k-1} \quad E\left[w^{T}_{k} \right] \right]^{T} \tag{5.69}$$

$$P^{a}_{k-1|k-1} = \begin{bmatrix} P_{k-1|k-1} & 0 \\ 0 & Q_k \end{bmatrix} \tag{5.70}$$

Then, $2L + 1$ sigma points are found through the following equations, where L is the dimension of the augmented state:

$$x^0_{k-1_{|k-1}} = x^a_{k-1_{|k-1}} \tag{5.71}$$

$$x^i_{k-1_{|k-1}} = \begin{cases} x^a_{k-1_{|k-1}} + (\sqrt{(L+\lambda)P^a_{k-1_{|k-1}}})_i & i = 1....L \\ x^a_{k-1_{|k-1}} + (\sqrt{(L+\lambda)P^a_{k-1_{|k-1}}})_{i-L} & i = L+1....2L \end{cases} \tag{5.72}$$

In equation (5.72), $(\sqrt{(L+\lambda)P^a_{k-1_{|k-1}}})_i$ is the i-th column of the matrix square root of $(L+\lambda)P^a_{k-1_{|k-1}}$. The sigma points are propagated through the non-linear function and the predicted state and covariance matrices are approximated using a weighted mean of the posterior sigma points from the previous time step:

$$x^i_{k_{|k-1}} = f(x^i_{k-1_{|k-1}}) \tag{5.73}$$

$$\hat{x}_{k_{|k-1}} = \sum_{i=0}^{2L} W_s^i x^i_{k_{|k-1}} \tag{5.74}$$

$$P_{k_{|k-1}} = \sum_{i=0}^{2L} W_c^i \left[x^i_{k_{|k-1}} - \hat{x}_{k_{|k-1}} \right] \left[x^i_{k_{|k-1}} - \hat{x}_{k_{|k-1}} \right]^T \tag{5.75}$$

The weights are given by:

$$W_s^0 = \frac{\lambda}{\lambda + L} \tag{5.76}$$

$$W_c^0 = \frac{\lambda}{\lambda + L} + (1 - \alpha^2 + \mu) \tag{5.77}$$

$$W_s^0 = W_c^i = \frac{1}{2(\lambda + L)} \tag{5.78}$$

$$\lambda = \alpha^2 (L + k) - L \tag{5.79}$$

where μ is related to the distribution of x, and α and κ are related to the spread of the sigma points. α is usually set to a small positive value (e.g., 0.001) and κ is usually set to zero [122]. In the update step, the predicted state and covariance matrices are augmented with the mean

and covariance of the measurement noise. Then, $2L+1$ sigma points are derived from the augmented state and covariance matrices:

$$x^a_{k_{|k-1}} = \left[\hat{x}^T_{k_{|k-1}} \quad E\left[v^T_k \right] \right]^T \tag{5.80}$$

$$P^a_{k_{|k-1}} = \begin{bmatrix} P_{k_{|k-1}} & 0 \\ 0 & R_k \end{bmatrix} \tag{5.81}$$

$$x^0_{k_{|k-1}} = x^a_{k_{|k-1}} \tag{5.82}$$

$$x^i_{k_{|k-1}} = \begin{cases} x^a_{k_{|k-1}} + (\sqrt{(L+\lambda)P^a_{k_{|k-1}}})_i & i = 1....L \\ x^a_{k_{|k-1}} + (\sqrt{(L+\lambda)P^a_{k_{|k-1}}})_{i-L} & i = L+1....2L \end{cases} \tag{5.83}$$

The sigma points are propagated through the non-linear observation function and the predicted measurement and measurement covariance is formed as the weighted mean of the sigma points:

$$\gamma^i_k = f(x^i_{k_{|k-1}}) \tag{5.84}$$

$$\hat{z}_{k_{|k-1}} = \sum_{i=0}^{2L} W^i_s \gamma^i_k \tag{5.85}$$

The predicted covariance and the state-measurement cross-covariance matrices are used to find the Kalman gain:

$$K_k = P_{x_k z_k} P^{-1}_{z_k z_k} \tag{5.86}$$

where:

$$P_{z_k z_k} = \sum_{i=0}^{2L} W_c{}^i \left[\gamma^i_k - \hat{z}_{k_{|k-1}} \right] \left[\gamma^i_k - \hat{z}_{k_{|k-1}} \right]^T \tag{5.87}$$

$$P_{x_k z_k} = \sum_{i=0}^{2L} W_c{}^i \left[x^i_{k_{|k-1}} - \hat{x}_{k_{|k-1}} \right] \left[\gamma^i_k - \hat{z}_{k_{|k-1}} \right]^T \tag{5.88}$$

Then, the state and covariance matrices are updated with the following equations:

$$\hat{x}_{k_{|k}} = \hat{x}_{k_{|k-1}} - K_k (z_k - \hat{z}_k) \tag{5.89}$$

$$P_{k_{|k}} = P_{k_{|k-1}} - K_k P_{z_k z_k} K^T_k \tag{5.90}$$

In the actual implementation of the UKF, there are two parameters which should be tuned carefully, the Q and R matrices. The measurement noise covariance matrix, R, is usually measured prior to the operation of the filter. In order to determine the measurement noise covariance, some offline samples of the measurements should be taken [119].

In [116], the UKF performance for DFIG parameter estimation was evaluated from different aspects: estimation accuracy, computation time, robustness to variation of the initial parameter estimates and filter gains.

The UKF was used to solve the parameter estimation problem for a wind-driven DFIG. The UKF estimation technique has significantly lower estimation errors and converges with fewer samples than the EKF method. Also, the study results suggested that UKF is more robust and better able to track parameter variations.

5.4.9 Extended unknown input observer-based on unscented transformation

The basic principle of unknown input observers (UIO) is to decouple disturbances from the state estimation error. In this method, the linear UIO design algorithm is extended to non-linear systems and, then, the observer gain is obtained using unscented transformation (UT). Motivated by these considerations, which use unscented transformation in the design procedure, UT has apparent advantages over linearisation algorithms and performs better than EKF [122]. The extended unknown input observers (EUIO) consist of first extending linear UIO to a non-linear framework and, then, employing UT to obtain the observer gain.

Consider the following non-linear discrete-time system with unknown input:

$$\begin{aligned} x_{k+1} &= g(x_k, u_k) + E_k d_k + R_k^1 f_k \\ y_{k+1} &= h(x_{k+1}) + R_k^2 f_k \end{aligned} \tag{5.91}$$

where d_k is an unknown input vector representing the disturbance, and E_k is α known matrix with appropriate dimensions corresponding to the state-space description of the linear, time-invariant system. The terms $R_k^1 f_k$ and $R_k^2 f_k$ represent the actuator and sensor fault, respectively.

Referring to [131], the EUIO estimate state is represented in equation (5.92):

$$\hat{x}_{k+1} = \overline{G}_k g(\hat{x}_k, u_k) + \overline{E}_k y_{k+1} + K_{k+1}(y_k - \hat{y}_k) \tag{5.92}$$

where $\overline{G}$ and $\overline{E}$ are matrices based on unscented transformation.

In general, faults can occur in actuators, components inside the system or sensors. To provide useful information for fault diagnosis, the residual should be defined such that it is approximately zero in a fault-free steady state and deviates from zero when a fault occurs:

$$\lim_{k\to\infty} r_k = \lim_{k\to\infty}(y_k - \hat{y}_k)\begin{cases} \leq T_k & = 0 \ \textit{fault free} \\ > T_k & \neq 0 \quad \textit{fault} \end{cases} \tag{5.93}$$

The residual is examined using a logical decision, with the aim of deciding if the fault has occurred when it is higher than a threshold (T_k) and avoiding wrong decisions such as a false alarm and missing a fault at less than T_k.

To evaluate the ability of the method, Zarei et al. (2010) [131] designed a single full-order observer to detect sensor faults in the presence of unknown inputs (disturbances). The goal was to discriminate between the fault effects and the effects of uncertain signals and perturbations. Simulation results demonstrated that it was possible to distinguish external disturbances from a response to a fault in the sensor.

5.4.10 Sequential Monte Carlo filtering-based method

The particle filter, a sequential Monte Carlo algorithm, was combined with the innovation-based fault detection techniques to develop a fault detection and isolation scheme in [128].

Consider that the system dynamics is known and given by:

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, w_{k-1}) \\ y_k &= h_k(x_k, v_k) \end{aligned} \tag{5.94}$$

Both the functions $f_{k-1}(\bullet)$ and $h_k(\bullet)$ can be non-linear or linear and are assumed to be known. The noise and disturbance are assumed to be additive and their characteristics known; when they are generally taken to be zero, this means Gaussian white noise.

Here, the type of faults of interest are the failure type, where the system parameter values jump to a new value reflected in a change in the functions $f_{k-1}(\bullet)$ and $h_k(\bullet)$. Such faults can be detected using the state observer approach or the filtering approach. The idea is to generate estimates of the states and the predicted outputs from these state estimates. The residuals from the output prediction are used in a measure which changes significantly under a failure type fault. Such a fault detection scheme facilitates online application since the state estimates and the predicted outputs can be generated online.

The approach is used specifically to replace, as the fault detection criteria, the EKF-based estimation scheme by the sequential Monte Carlo filter, and the weighted sum squared residual measure by an appropriate innovations likelihood measure. The likelihood is a model or hypothesis probability which is useful in model comparison. False alarms may occur with this criterion if it is based on a single output measurement. A robust criterion is obtained by determining the likelihood over a window of length *K*, leading to a detection criterion which makes use of the complete probability density function state information given by the sequential Monte Carlo filter.

I- Fault Detection

The following algorithm describes the complete sequential Monte Carlo filter-based fault detection scheme:

1. State prediction:

Samples $x_k^*(i)$ are generated as in the sequential Monte Carlo filter prediction step [40]; where $i = 1, 2, \ldots, n$.

2. Output prediction:

Output prediction samples $y_k^*(i)$ are generated using the measurement equation (5.94):

$$y_k^*(i) = h_k(x_k^*(i)) \qquad (5.95)$$

3. Residual generation:

The sample mean of the predicted measurements is computed:

$$r_k = y_k - y_k^*(i) \tag{5.96}$$

4. Fault detection:

The innovation likelihood is given by:

$$p(r_{k|z_k}) = \frac{1}{N}\sum_{i=1}^{N} \tilde{q}_k(i) \tag{5.97}$$

where z_k denotes the set of measurements up to time *k*, ie., $z_k = y_1, y2,\ldots, y_k$. $\tilde{q}_k(i)$ are the un-normalised weights:

$$\tilde{q}_k(i) = p(y_{k|x_k^*(i)}) \tag{5.98}$$

The log likelihood is computed as in the equation(5.99):

$$\mathcal{L}(k) = \sum_{j=k-k+1}^{k} -\ln(p(r_{k|z_k})) \tag{5.99}$$

The condition $\mathcal{L}(k) > \epsilon$ is examined for the presence of a fault.

5. State update:

Weights $\tilde{q}_k(i)$ for the samples $x_k^*(i)$ are generated as in the sequential Monte Carlo filter update step.

6. Resample:

Samples $x_k(i)$ are obtained from re-sampling, as in the sequential Monte Carlo filter resample step. Then, the steps are repeated recursively for each *k.*

II- Fault Isolation

The scheme outlined above is a fault detection scheme which cannot be extended readily to fault isolation. One approach to fault isolation is to estimate the parameters of the model and track the changes in their values. Such a procedure requires the simultaneous estimation of the states and the parameters; this can be achieved using the augmented states model [129]. Recently, it was proposed to use a sequential Monte Carlo filter for estimating the states and the parameters [130].

The idea is to use an augmented state $\zeta^T = [x^T, \theta^T]$ and rewrite the state-space model in terms of ζ . The following set of equations results:

$$\begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} f_{k-1}(x_{k-1}, \theta_{k-1}, w_{k-1}) \\ \theta_{k-1} + w'_{k-1} \end{bmatrix} \qquad (5.100)$$
$$y_k = h_k(x_k, \theta_k, v_k)$$

where the dependency of the functions $f_{k-1}(\bullet)$ and $h_k(\bullet)$ on the parameters θ are made explicit. The disturbance term w'_{k-1} is introduced using a random walk model for parameter evolution to allow the exploration of the parameter space, as is done typically.

From the state-space equation (5.100), the sequential Monte Carlo filter can be used to estimate the states and the parameters. Referring to [128], an estimate can be written as in equation (5.101):

$$\hat{\zeta}_k = \sum_{i=1}^{N} q_k(i) \zeta_k^*(i) \qquad (5.101)$$

The estimate is, in effect, a weighted average of the particles or samples representing the basic distribution. The parameter estimates $\hat{\theta}_k$ can be compared to the nominal values θ_0 , as a means for fault detection and its deviation $\tilde{\theta}_k$ in equation (5.102) can be used for fault isolation.

$$\tilde{\theta}_k = \theta_0 - \hat{\theta}_k \qquad (5.102)$$

The augmented state-space model is attractive in principle. However, this increases the dimensionality of the model and, thereby demands an increase in the number of particles for sufficiently accurate results.

A sequential Monte Carlo filter-based approach to a fault detection and isolation scheme was developed for a general non-linear system with non-Gaussian noise and disturbances.

According to Kadirkamanathan et al. (2000) [128], the detectability of the sequential Monte Carlo filtering approach is superior to the EKF-based scheme, especially in the case where the system model is highly non-linear. The fault isolation scheme is shown, also, to identify the parameter related with the fault and the level of the fault.

5.4.11 Summary

This section presented a survey of methodologies in the theory of a non-linear observer-based fault diagnosis. Because of the existing volume of research activity in this field, it was impossible to include a complete representation of the methodologies. Therefore, a focus was given to those methods which can be applied for monitoring wind turbines and which may gain some relevance for future research and practical applications (see Table 5.1). The methods, discussed in the survey, give rise to great encouragement and may help to motivate intensive future efforts towards the practical application of these ideas.

Table 5.1: Comparing the methodologies of non-linear observer in section 5.4

Methodology	Fault detection	Fault isolation	Application for wind turbines	
Lyapunov-based method	Yes	Yes	No	sensor faults for a synchronous machine
Tsinias observer	Yes	No	No	actuator sensor component faults
Thau non-linear observer	Yes	No	No	fault-tolerant control
Extended Luenberger observer	Yes	No	No	control systems' feedback sensors
State-dependent non-linear observer	No	No	No	control design
Construct the non-linearity from direct measurements	Yes	Yes	Yes	wind turbine generator
Extended Kalman filter	No	No	Yes	parameter estimation for a DFIG
Unscented Kalman filter	No	No	Yes	parameter estimation for a DFIG
Extended unknown Input observer	Yes	No	No	sensor faults
Sequential Monte Carlo filtering-based method	Yes	Yes	No	general non-linear system with non-Gaussian noise and disturbances

5.5 Design of state-dependent non-linear observer for fault detection and isolation

A non-linear observer is a dynamic filter that estimates the states or outputs of a system based on a mathematical model, sensor measurements and input commands. In the previous chapter, a linear observer was designed and applied for the case where the rotor speed varies slowly. However, in the real case, both the wind speed and the rotor speed of the generator are variable, resulting in non-linear dynamics which must be considered in the FDI design. Therefore, it is necessary to develop a non-linear observer.

This section presents a framework for designing a non-linear observer using the state-dependent differential Riccati equation (SDDRE) rather than the algebraic Riccati equations, which may pose an overly restrictive requirement on the observability and controllability of the system. The SDRE, based on an extended linearisation of the process dynamics and a solution of the SDDRE, can be determined using numerical integration methods. Matlab was used to solve problems for ordinary differential equations; specifically the solver *ode45* was used. The *ode45* solver is based on an explicit Runge-Kutta formula [132].

The work presented in this section aims at detecting faults in the sensors of a wind turbine; such faults are related to pitch angle, rotor angle, generator rotor angle, wind turbine rotor speed, generator rotor speed, electric torque and wind turbine torque.

5.5.1 Non-linear model for 5MW wind turbine

Based on the wind turbine equations (3.3), (4.35) and (4.36), the differential equations for the subsystems are:

- ***Actuator model :***

$$\dot{\beta} = \frac{1}{\tau_\beta}(\beta_d - \beta)$$

- ***Drive-train model:***

$$\dot{\omega}_{wt} = \frac{1}{J_T}(T_{wt} - K_s\theta_K - C_s\omega_{wt} + \frac{C_s}{n_g}\omega_m)$$
$$\dot{\omega}_m = \frac{1}{JG}(\frac{K_s}{n_g}\theta_K + \frac{C_s}{n_g}\omega_{wt} - \frac{C_s}{n_g^2}\omega_m - T_e^c - b_d\omega_m)$$

- ***Doubly-fed induction generator model***:

$$\sigma L_r \frac{di_{dr}}{dt} = -R_r i_{dr} + (\omega_s - \omega_m)\sigma L_r i_{qr} + v_{dr}$$
$$\sigma L_r \frac{di_{qr}}{dt} = -R_r i_{qr} - (\omega_s - \omega_m)(\sigma L_r i_{dr} + \frac{\sqrt{3}L_m V_s}{\omega_s L_s}) + v_{qr}$$

where: $v_{ds} \approx 0,\ v_{qs} = |\vec{v}_s|$

The non-linear model of a 5MW wind turbine can now be written as:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned} \tag{5.103}$$

where

$$\dot{x} = \begin{bmatrix} -\frac{1}{\tau_\beta}x_1 + \frac{1}{\tau_\beta}u_1 \\ x_3 - \frac{1}{n_g}x_4 \\ -\frac{K_s}{J_T}x_2 - \frac{K_s}{J_T}x_3 + \frac{C_s}{J_T n_g}x_4 + \frac{1}{J_T}u_2 \\ \frac{K_s}{J_G n_g}x_2 + \frac{C_s}{J_G n_g}x_3 - (\frac{C_s}{J_G n_g^2} + b_d)x_4 - \frac{1}{J_G}u_3 \\ -x_6 x_4 - \frac{R_r}{\sigma L_r}x_5 + \omega_s x_6 + \frac{1}{\sigma L_r}u_4 \\ \frac{\sqrt{3}L_m V_s}{\sigma L_s L_r \omega_s}x_4 + (x_4 - \omega_s)x_5 - \frac{R_r}{\sigma L_r}x_6 + \frac{1}{\sigma L_r}u_5 - \frac{\sqrt{3}L_m V_s}{\sigma L_s L_r} \end{bmatrix} \tag{5.104}$$

and the output equation is given by:

$$y=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ K_s x_2 + C_s x_3 - \dfrac{C_s}{n_g} x_4 \\ \dfrac{n_p \sqrt{3} L_m V_s}{L_s \omega_s} \end{bmatrix} \tag{5.105}$$

The state vector, input vector and output vector are defined as:

$$x=\begin{bmatrix} \beta \\ \theta_k \\ \omega_{wt} \\ \omega_m \\ i_{dr} \\ i_{qr} \end{bmatrix}, \qquad u=\begin{bmatrix} \beta_d \\ T_{wt} \\ T_e^c \\ v_{dr} \\ v_{qr} \end{bmatrix}, \qquad y=\begin{bmatrix} \beta \\ \theta_k \\ \omega_{wt} \\ \omega_m \\ T_{wt} \\ T_e \end{bmatrix}$$

5.5.2 Non-linear observer design

The non-linear dynamics model given in equations (5.104) and (5.105) can be represented by the following linear structure with state-dependent coefficients (SDC):

$$\begin{aligned} \dot{x} &= A(x)x + B(u)u \\ y &= Cx + Du \end{aligned} \tag{5.106}$$

where:

$$A(x)=\begin{bmatrix} -\frac{1}{\tau_\beta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{n_g} & 0 & 0 \\ 0 & -\frac{K_s}{J_T} & -\frac{K_s}{J_T} & \frac{C_s}{J_T n_g} & 0 & 0 \\ 0 & \frac{K_s}{J_G n_g} & \frac{C_s}{J_G n_g} & -(\frac{C_s}{J_g n_g^2}+b_d) & 0 & 0 \\ 0 & 0 & 0 & \text{-x}_6 & -\frac{R_r}{\sigma L_r} & \omega_s \\ 0 & 0 & 0 & \frac{\sqrt{3}L_m V_s}{\sigma L_s L_r \omega_s} & (\text{x}_4-\omega_s) & -\frac{R_r}{\sigma L_r} \end{bmatrix} \quad (5.107)$$

$$B(u)=\begin{bmatrix} \frac{1}{\tau_\beta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_T} & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{J_G} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma L_r} & 0 \\ 0 & 0 & 0 & 0 & (\frac{1}{\sigma L_r}-\frac{\sqrt{3}L_m V_s}{\sigma L_s L_r u_5}) \end{bmatrix} \quad (5.108)$$

$$C=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & K_s & C_s & -\frac{C_s}{n_g} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{n_p\sqrt{3}L_m V_s}{L_s \omega_s} \end{bmatrix} \quad (5.109)$$

After bringing the system to the SDC form, as in equation (5.106), an observer with the same structure as in equation (5.106) was constructed, with the addition of the driving feedback term whose role is to reduce the observation error to zero as:

$$\dot{\hat{x}} = A(\hat{x})\hat{x} + B(u)u + K(\hat{x})[y - C\hat{x}] \tag{5.110}$$

The non-linear observer gain matrix in equation (5.46) ($K(\hat{x}) = P(\hat{x})\ C^T R^{-1}$) is solved using the Riccati differential equation (5.48) to find the matrix $P(\hat{x})$ using $A(x)$, C and $Q = C^T C \geq 0$, and $R(x) > 0$ for all x.

5.5.3 Non-linear observer examination

The performance of a non-linear observer was investigated in comparison with the output of the wind turbine using simulations with effective wind speeds of 15 m/s and 7 m/s (such wind speeds produce nominal and less than nominal power, respectively); process disturbances and sensor noise were not considered in the simulation. The value of $R=10^7$ is that which gives a fast response and has less error in the steady-state period.

Figure 5.2 and Figure 5.3 show the simulation results for $R=10^7$, comparing system outputs and estimated states. The sub-figures confirm the good performance of the non-linear observer for faults relating to (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle (θ_K), (c) wind turbine rotor speed, (d) generator rotor speed, (e) electric torque and (f) wind turbine torque with effective wind speeds of 7 and 15 m/s, respectively.

Estimated states values converge rapidly to the non-linear model output. Therefore, it can be concluded that the non-linear observer matches well to the estimated states of the wind turbine's mechanical parts.

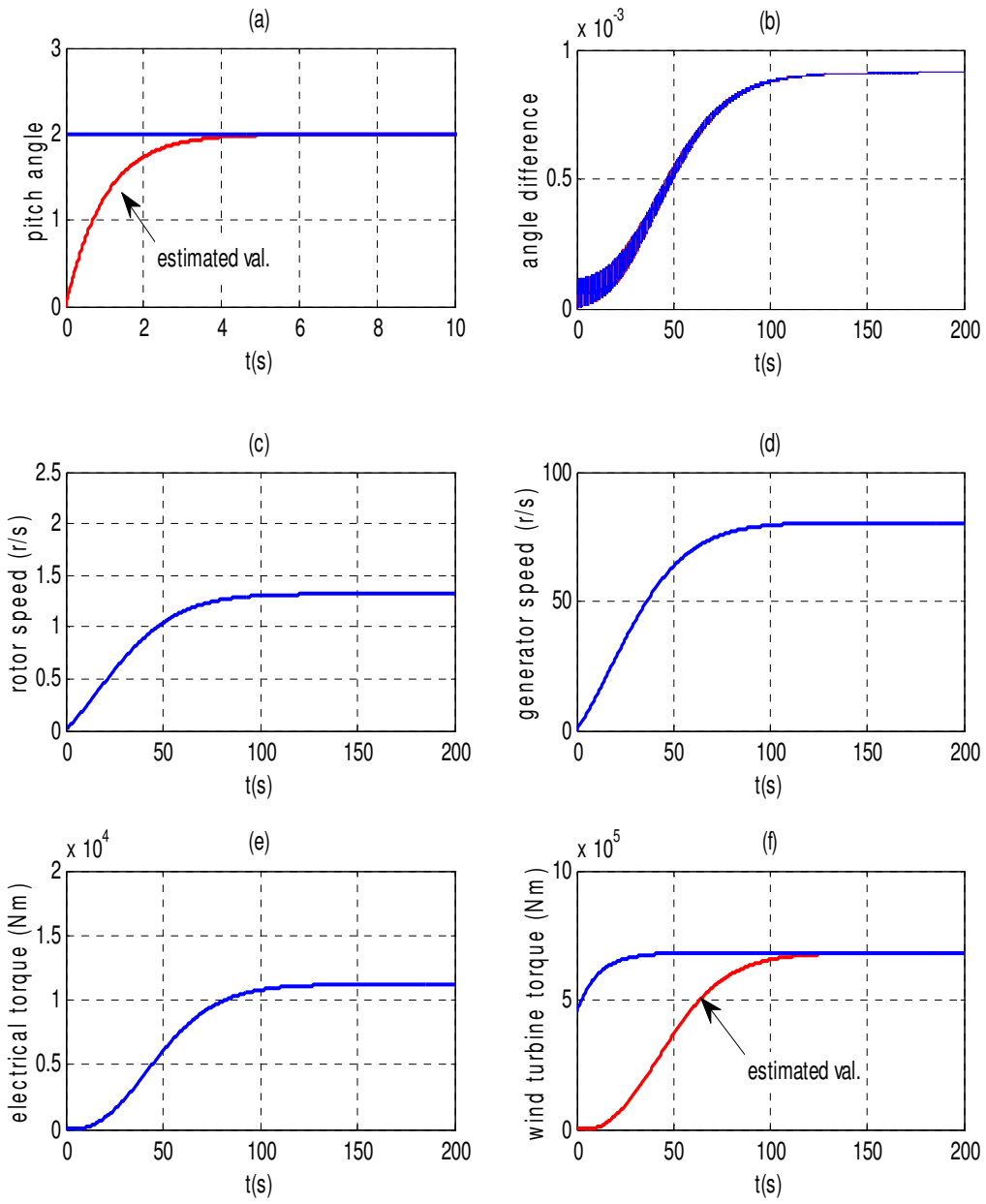


Figure 5.2: Comparison of system outputs with non-linear observer estimation of faults related to (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle (θ_K), (c) wind turbine rotor speed, (d) generator rotor speed, (e) electric torque; and (f) wind turbine torque with an effective wind speed of 7 m/s. The outputs are similar in the sub-figures (b, c, d and e)

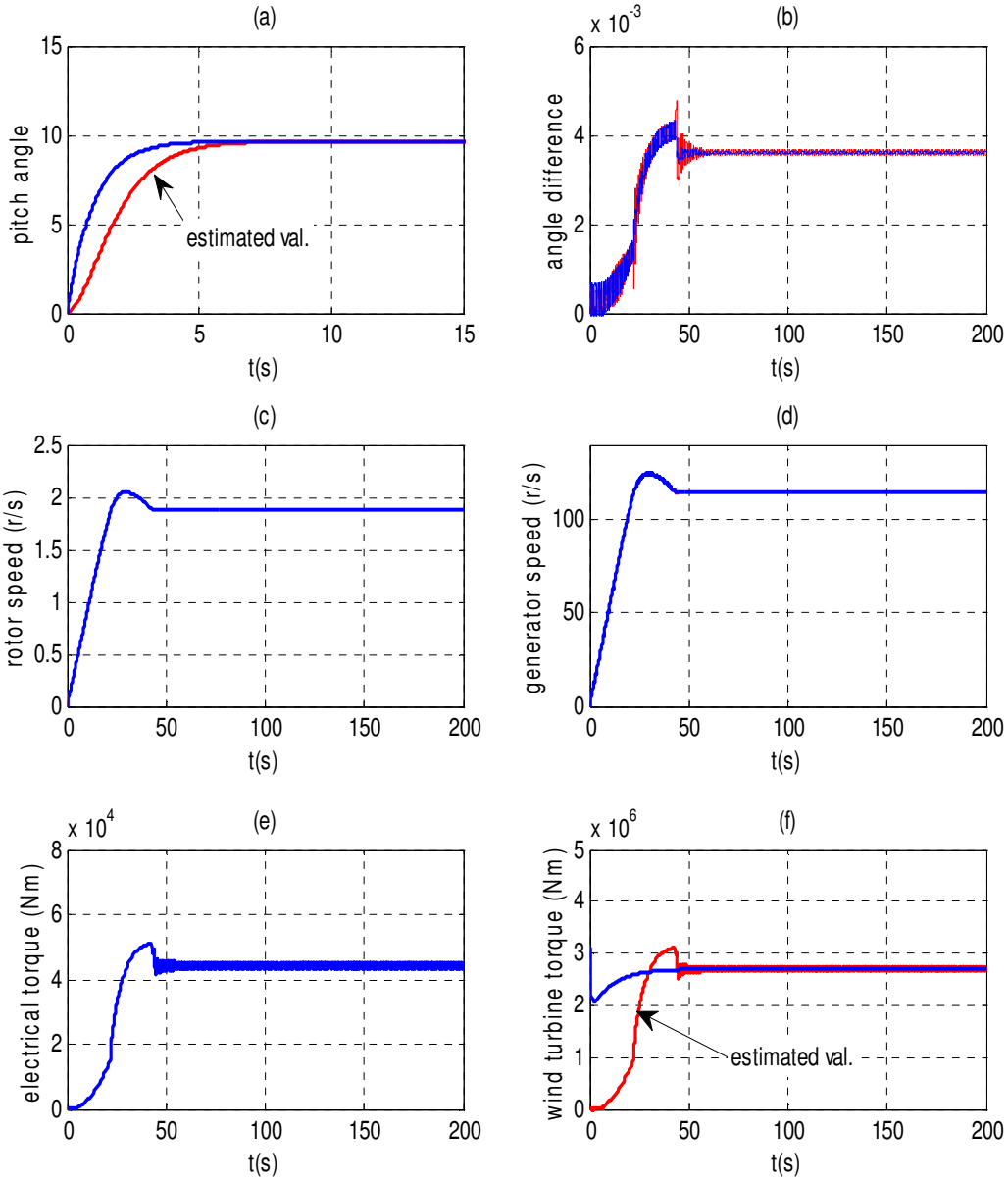


Figure 5.3: Comparison of system outputs with non-linear observer estimation of faults related to (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle (θ_K), (c) wind turbine rotor speed, (d) generator rotor speed, (e) electric torque, and (f) wind turbine torque, with an effective wind speed of 15 m/s. The outputs are similar in the sub-figures (b, c, d and e)

5.5.4 Stability of state-dependent observer

SDRE stabilisation refers to the use of the state-dependent Riccati equations to construct a non-linear observer for non-linear systems. The main concept is to represent the non-linear system in the form:

$$\dot{x}=A(x)x+B(u)u$$

and to use the feedback gain:

$$K(\hat{x})=P(\hat{x})C^{T}R^{-1}$$

where *P(x)* is obtained from the SDDRE:

$$\dot{P}(\hat{x})=P(\hat{x})A^{T}(\hat{x})+A(\hat{x})P(\hat{x})-P(\hat{x})C^{T}R^{-1}CP(\hat{x})+Q$$

and Q and R are design parameters that satisfy the point-wise positive definiteness condition:

$$Q = C^{T}C \geq 0 \text{ , and } R > 0 \text{ for all x}$$

To define the error, equation (5.110) is subtracted from equation (5.106):

$$\begin{aligned} \dot{x}-\dot{\hat{x}} &= A(x-\hat{x})-P(\hat{x})C^{T}R^{-1}(y-\hat{y}) \\ &= [A(x)-K(\hat{x})\ C]\ (x-\hat{x}) \end{aligned} \qquad (5.111)$$

Define the difference between $x(t)$ and $\hat{x}(t)$ as the state error vector, *e*, thus the dynamic error can be written as:

$$\dot{e}=\dot{x}-\dot{\hat{x}}=[A(x)-P(\hat{x})\ C^{T}R^{-1}C]\ e(t) \qquad (5.112)$$

Equation (5.112) illustrates the dynamic behaviour of the innovation signal and this is governed by the eigenvalues of the matrix $[A(x)-P(\hat{x})\ C^{T}R^{-1}C]$.

If the eigenvalues of the matrix $[A(x)-P(\hat{x})\ C^{T}R^{-1}C]$ are in the left half-plane, then the system is stable. The simulation results proved that the state-dependent non-linear observer is stable at all the tested points.

5.5.5 Non-linear observer-based FDI scheme

The state-space model of the wind turbine in equation (5.106) can be extended to include faults and disturbances as follows:

$$\begin{aligned} \dot{x} &= A(x)x + B(u)u + R_1 f + d \\ y &= Cx + Du + R_2 f \end{aligned} \tag{5.113}$$

The equation of the observer using this model for $d = f = 0$, will be:

$$\begin{aligned} \dot{\hat{x}} &= A(\hat{x})\hat{x} + B(u)u + K(\hat{x})[y - C\hat{x}] \\ \hat{y} &= C\hat{x} + Du \end{aligned} \tag{5.114}$$

where f is the fault vector, which is considered to be an unknown time function. The vector $d(x)$ is the disturbance vector, which can be written as:

$$d = \Delta A(x)x + \Delta B(u)u \tag{5.115}$$

The residual generator, studied in this section, is based on a non-linear observer; it is employed to the monitoring system represented by equation (5.113). To define the state error, equation (5.114) is subtracted from equation (5.113) to obtain the estimated error equation:

$$\dot{e} = \dot{x} - \dot{\hat{x}} = (A(x) - P(\hat{x})C^T R^{-1} C)e + d + R_1 f - P(\hat{x})C^T R^{-1} R_2 f \tag{5.116}$$

and the residual equation:

$$r = Q[y - \hat{y}] = Q[C(x - \hat{x}) + R_2 f] = Q\,[Ce + R_2 f] \tag{5.117}$$

To implement this approach, two steps should be taken. Firstly, the sensitive and insensitive relationships between residuals and faults should be defined for each type of fault by rewriting equation (5.113) in the form:

$$\begin{aligned} \dot{x} &= A(x)x + B(u)u + R_1^a f^a + R_1^s f^s + R_1^c f^c + d \\ y &= Cx + Du + R_2^a f^a + R_2^s f^s + R_2^c f^c \end{aligned} \tag{5.118}$$

where R_1 and R_2 are fault matrices and *a, s* and *c* denote actuator, sensor and component, respectively.

Secondly, a set of residuals should be designed according to equation (5.118) and as they are presented in equations (5.119) - (5.121). Faults are classified into three groups: actuator, sensor and component faults. The performance indices, given in the robust residual generation described in Chapter 4, are revised as follows:

$$J_{af}(P,R,Q_a) = \sup_{\omega\in[\omega_1,\omega_2]} \bar{\sigma}\{-[Q_aD + Q_aC(j\omega I - A(x) + P(x)C^TR^{-1}C)^{-1} (B(u) - P(x)C^TR^{-1}D)]\ \} \tag{5.119}$$

$$J_{sf}(P,R,Q_s) = \sup_{\omega\in[\omega_1,\omega_2]} \bar{\sigma}\{[Q_sI + Q_sC(j\omega I - A(x) + P(x)C^TR^{-1}C)^{-1} (-P(x)C^TR^{-1}I)]^{-1}\} \tag{5.120}$$

$$J_{cf}(P,R,Q_c) = \sup_{\omega\in[\omega_1,\omega_2]} \bar{\sigma}\{-[Q_cC(j\omega I - A(x) + P(x)C^TR^{-1}C)^{-1})]\} \tag{5.121}$$

$$J_d(P,R) = \| (A(x) - P(x)C^TR^{-1}C)^{-1} \| \tag{5.122}$$

The actuator, sensor and component fault matrices are defined as:

$$R_1 = \begin{cases} 0 & sensor\ fault \\ B(u) & actuator\ fault \\ A(x) & component\ fault \end{cases} \tag{5.123}$$

$$R_2 = \begin{cases} C & sensor\ fault \\ D & actuator\ fault \\ 0 & component\ fault \end{cases} \tag{5.124}$$

From equations (5.120) and (5.122), it can be seen that the performance indices $J_{sf}(P(x),R,Q_s)$ and $J_d(P(x),R)$ are functions of *P(x), R* and Q_s. Therefore, these parameters should be optimised for observer stability by maximising the effect of sensor faults and minimising the effect of the disturbance [91]; this is a multi-objective optimisation problem. Firstly, *P(x)* and *R*, which are represented in the Riccati equation (5.48), are optimised to find the best non-linear observer gain matrix $K(\hat{x})$. The initial matrix P_0 is assumed to be zero and $\dot{P}(x)$ is:

$$\dot{P}(x)=\begin{bmatrix} \dot{P}_{1,1}(x) & \cdots & \dot{P}_{1,5}(x) & \dot{P}_{1,6}(x) \\ \dot{P}_{2,1}(x) & \cdots & \cdots & \dot{P}_{2,6}(x) \\ \vdots & \cdots & \cdots & \vdots \\ \dot{P}_{i,1}(x) & \cdots & \cdots & \dot{P}_{6,6}(x) \end{bmatrix} \tag{5.125}$$

From equations (5.120) and (5.122) it can be seen that the norm of the performance index $J_{sf}(P,R,Q_s)$ is very large and tends to infinity (see the procedure to minimise two objective functions for sensor faults and MOGA, presented in sections 5.4.4.1 and 4.5.4.2). Therefore, only the disturbance effects $J_d(P,R)$ need to be optimised. From equation (5.48) it can be seen that *P* is a function of states. Consequently, only *R* is optimised, giving better sensitivity to faults.

For instance, If a small value is selected for *R*, the non-linear observer is insensitive to faults, as shown in Figure 5.4, where |system output+fault| ≈ |estimated value|. Hence, a value for *R* should be selected very high that gives better sensitivity to faults (form siumlatin the best value is $R=10^7$), as shown in Figure 5.5, where |system output+fault| > |estimated value|.

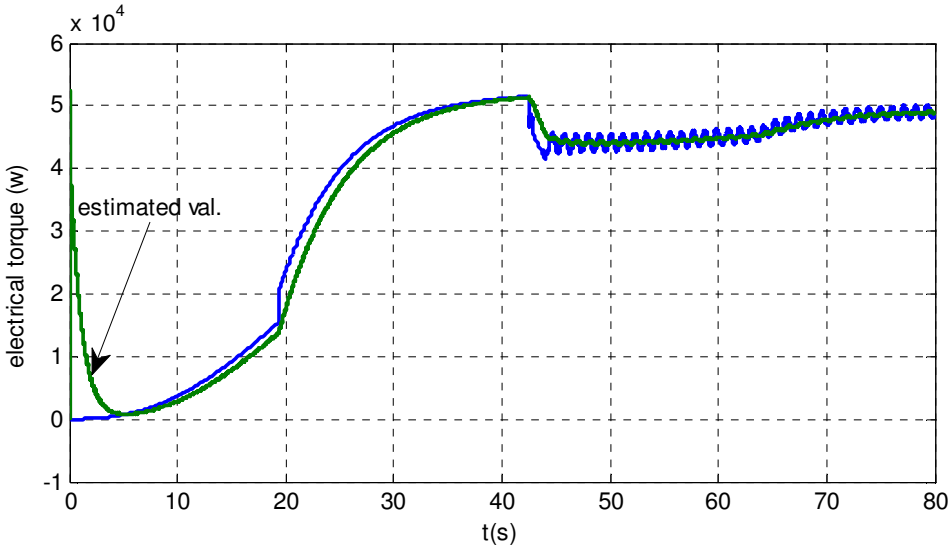


Figure 5.4: Comparison of system outputs with a non-linear observer where 10% electric torque sensor fault occurs at time-point t=45 s with an effective wind speed of 13 m/s and R=1

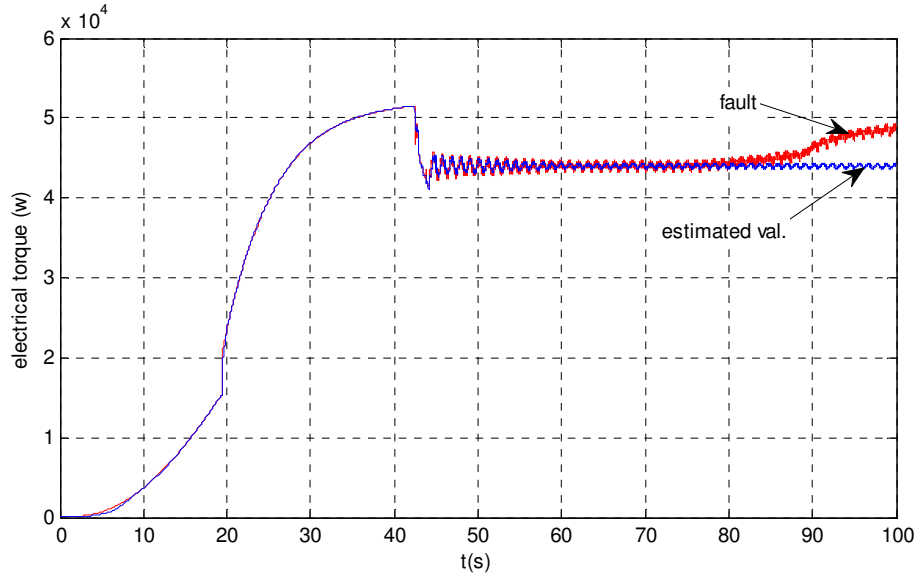


Figure 5.5: Comparison of system outputs with a non-linear observer where 10% electric torque sensor fault occurs at time-point t=70 s with an effective wind speed of 13 m/s and R=10^7

The residual weighting factor matrix Q_{sk} is defined as that which gives the appropriate residual dimensions for all output by using the Simulink model in Matlab. The values of Q_{sk} are defined in equation (5.126).

$$Q_{sk} = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 \end{bmatrix} \tag{5.126}$$

where

$$q_1 = 10^5,\ q_2 = 10^5,\ q_3 = 10^5,\ q_4 = 5*10^2,\ q_5 = 10,\ q_6 = 1.$$

5.5.6 Non-linear observer-based sensor fault detection and isolation scheme

To simplify the steps of designing robust sensor-based FDI, we assume all actuators and components are fault free. Equation (5.113) can then be simplifed to:

$$\begin{aligned} \dot{x} &= A(x)x + B(u)u + d \\ y &= Cx + R_2 f \end{aligned} \tag{5.127}$$

The residual generator for each sensor will be:

$$r_k = Q_{sk}[(Ce + R_2 f(C_m - C_k)] \tag{5.128}$$

where k is the number of the measurement sensor, $C_k \in R^{pxp}$ is obtained from the matrix C by assuming the k_{th} row equals zero and C_m is equal to the matrix C. From equation (5.128) it is obvious that each residual generator is driven, so that all other residuals equal zero. From the above, a set of robust and observer-based sensor-fault isolation schemes were designed, as illustrated in Figure 5.6.

Each sensor residual (r_k) is separated from the output of the residual (r) by $(C_m - C_k)$ and the dimension of r_k is modified using Q_{sk}. Compared to other approaches, this approach uses a bank of observers such as a structured residual set designed by a dedicated or a generalised observer scheme; this approach is advantageous since it uses only a single observer [91].

In practice, it was insufficient to identify faults based on residual curves because the system was non-linear, and it worked at different operating points; it was necessary to design a dynamic threshold for each sensor taking into account individual sensor accuracy, supplied by the manufacturer. Therefore, residuals were tested using an interval test dynamic threshold (see Figure 5.6). This test gave an output of one (no sensor faults occurred) if the input of the measurement output signals did not exceed the absolute value of the sum of the estimated value, sensor accuracy and sensor noises. The test gave an output of zero (case of sensor fault) if the input of the measurement output signal exceeded the absolute value of the sum of the

estimated value, sensor accuracy and sensor noises. The mathematical model of an interval test dynamic block output is based on equation (5.129):

$$\begin{aligned} y_k(t) &\le \left|\hat{y}_k(t) + \delta y_k(t) + \eta_k\right| \Rightarrow f_k(t) = 1 \text{ fault free} \\ y_k(t) &> \left|\hat{y}_k(t) + \delta y_k(t) + \eta_k\right| \Rightarrow f_k(t) = 0 \text{ fault} \end{aligned} \tag{5.129}$$

where $y_k(t)$ is the measurement output signal, $\hat{y}_k(t)$ is the estimated value, η_k is the sensor accuracy and $\delta y_k(t)$ is the sensor noise.

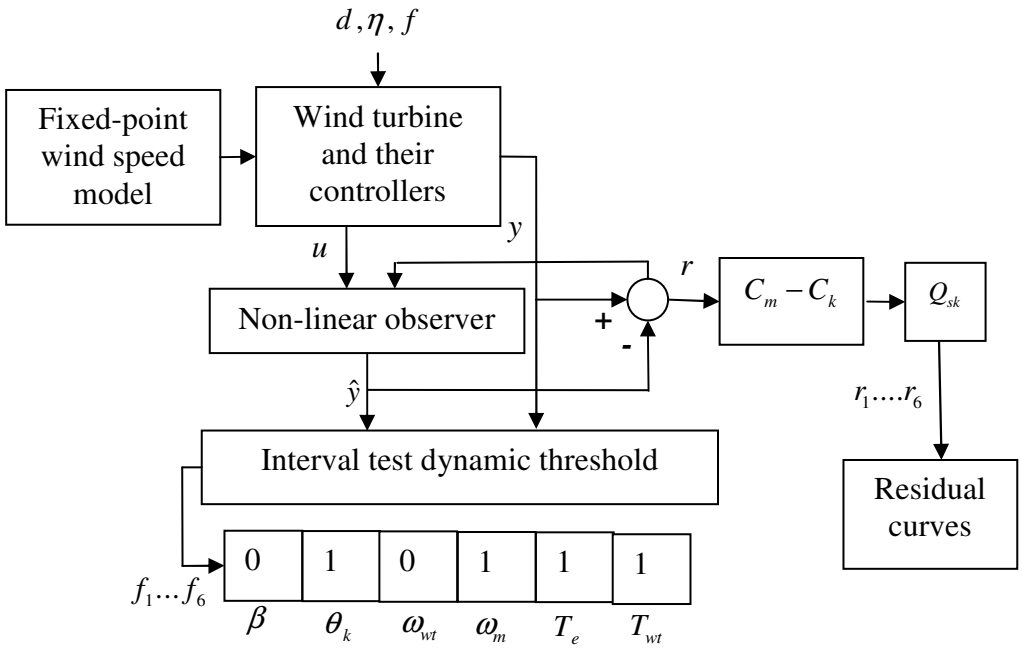


Figure 5.6: Robust non-linear observer-based fault detection and isolation scheme. The Simulink model is illustrated in Figures B.19. d, η and f are disturbance, sensor noise and sensor faults, respectively. Here, "0" is indicated on the display when pitch angle and turbine speed sensor faults occur.

5.5.7 Simulation results

A non-linear observer-based residual generator was designed, as shown in Figure 5.6 and Appendix B.19. In order to assess the performance of the method, different types of additive sensor fault were applied. The fault types were pitch angle, the difference between turbine rotor angle and generator rotor angle, wind turbine rotor speed, generator rotor speed, electric torque and wind turbine torque.

The simulation was repeated for two scenarios, illustrated in Table 5.2. In the first scenario, it was assumed that only one fault occurs at time-point t=150 s, with an effective wind speed of 10 m/s. In the second scenario, a number of faults occurred each at different time-points, with an effective wind speed of 12 m/s. In all simulation cases, disturbance and sensor noises were assumed to be present.

Under normal operation, when the wind turbine had only disturbance and noise signals, the average of the residual norms was very small (≈ 0), as shown in Figure 5.7. From Figures 5.8 - 5.11, which show the residual norms, when a fault occurred in β , θ_k, ω_{wt} or ω_m respectively, only the related residual increased, which led to fast and easy fault location. In the case where there was a T_{wt} or T_e fault, as shown in Figures 5.12 and 5.13, it can be seen that the response of the fault detection was very fast in the case of wind turbine torque fault and slow (delay time ≈10 s) in the case of an electric torque fault; this allows both faults to be identified and located. This is further demonstrated by the behaviour of the residual curves in Figure 5.14, where both wind turbine and electric torque faults occurred at the same time. Here, first the turbine rotor torque sensor becomes apparent and, then, after a few seconds, the electric torque, with its slow response, becomes apparent.

In the case of multi-malfunction, the result of the simulation demonstrated that detection and isolation is achieved, as illustrated in Figure 5.15, when more than one sensor fault occurred ($10\%\beta$, $10\%\theta_k$, $10\%\omega_{wt}$, $9\%\omega_m$ and $10\%T_e$ at the time-points t=250, 220, 200 180 and 150 s, respectively for an effective wind speed of 12 m/s). For example, Figure 5.6 clearly illustrates the robustness of this method when

compared to FDI. It demonstrates that as pitch angle and rotor speed sensor faults occur, they are represented as zeroes in the fault display, as shown in Figure 5.6. By inspecting the residual curves, it is possible to identify the start time for the pitch angle and rotor speed sensor faults.

Table 5.2: Sensor noise and percent of faults (system disturbance = 0.01)

	β	θ_K	ω_{wt}	ω_m	T_{wt}	T_e
Sensor noise	0.0001	0.0001	0.0001	0.05	0.1	0.1
First simulation scenario (one fault occurs at t=150 s and v=10m/s)	5%	0	0	0	0	0
	0	8%	0	0	0	0
	0	0	8%	0	0	0
	0	0	0	8%	0	0
	0	0	0	0	8%	0
	0	0	0	0	0	8%
Second simulation scenario (multiple faults and v=12m/s)	0	0	0	0	8% at 150s	10% at 150s
	10% at 250s	10% at 220s	10% at 200s	9% at 180s	0	10% at 150s

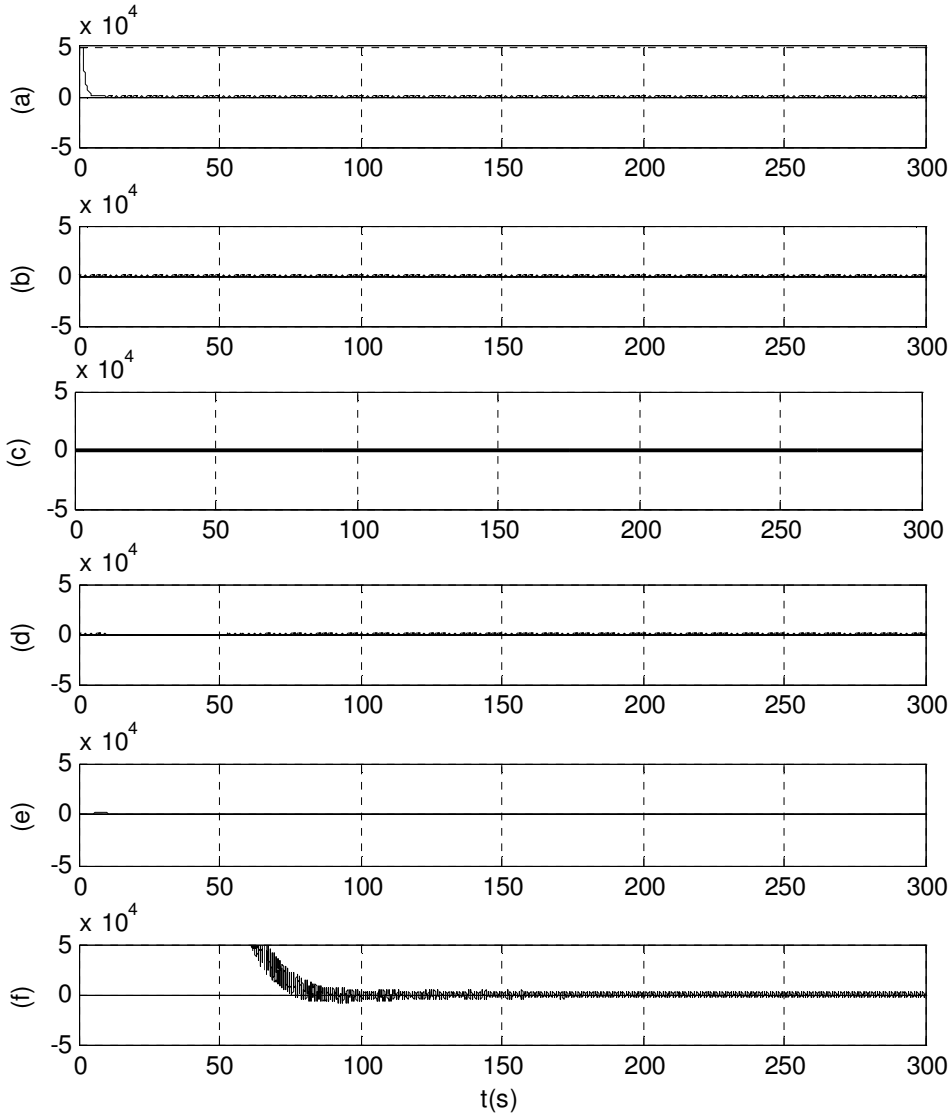


Figure 5.7: Residual norms in the case of no fault for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e) electric torque and (f) wind turbine torque (f)

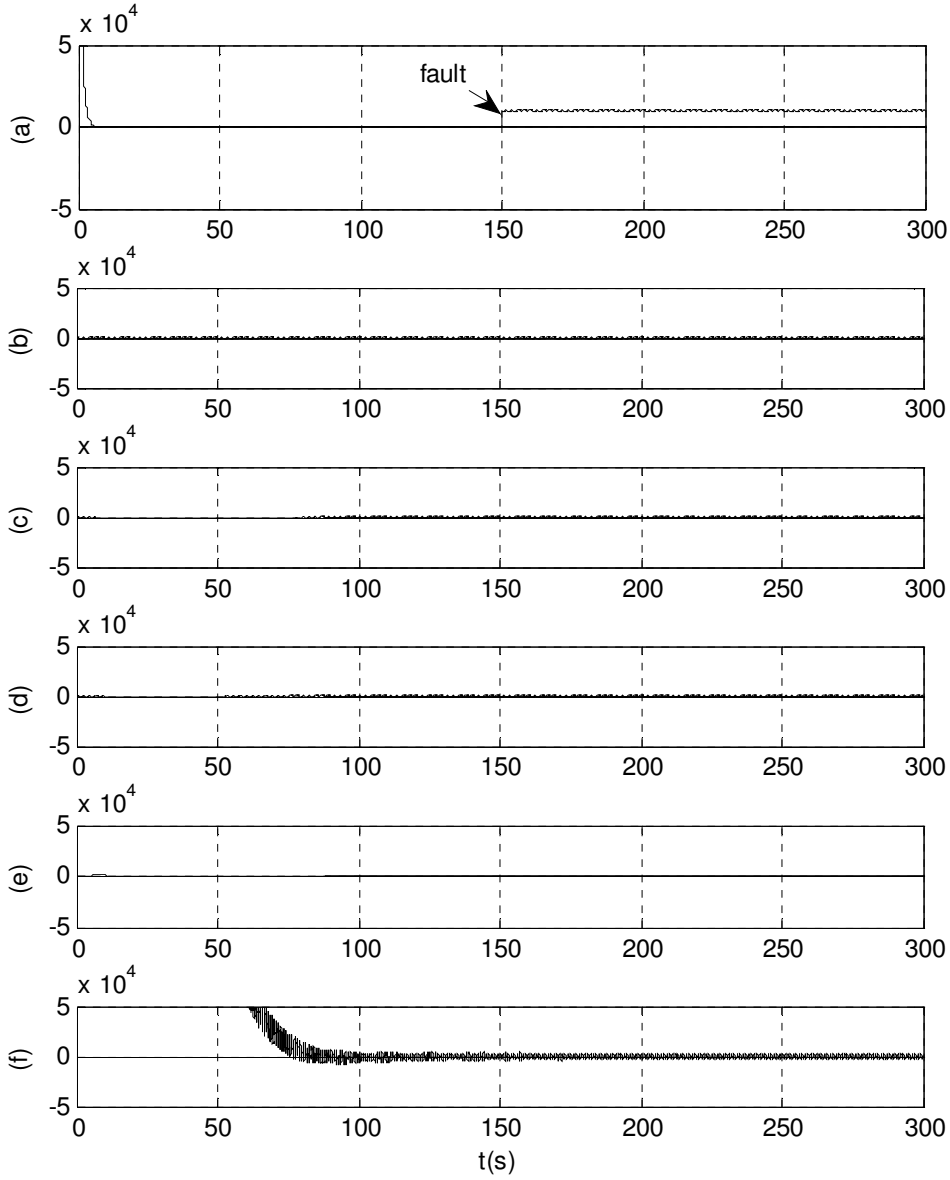


Figure 5.8: Residual norms in the case of a 5% pitch angle sensor fault at time-point t=150 s and an effective wind speed of 10 m/s for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e) electric torque and (f) wind turbine torque

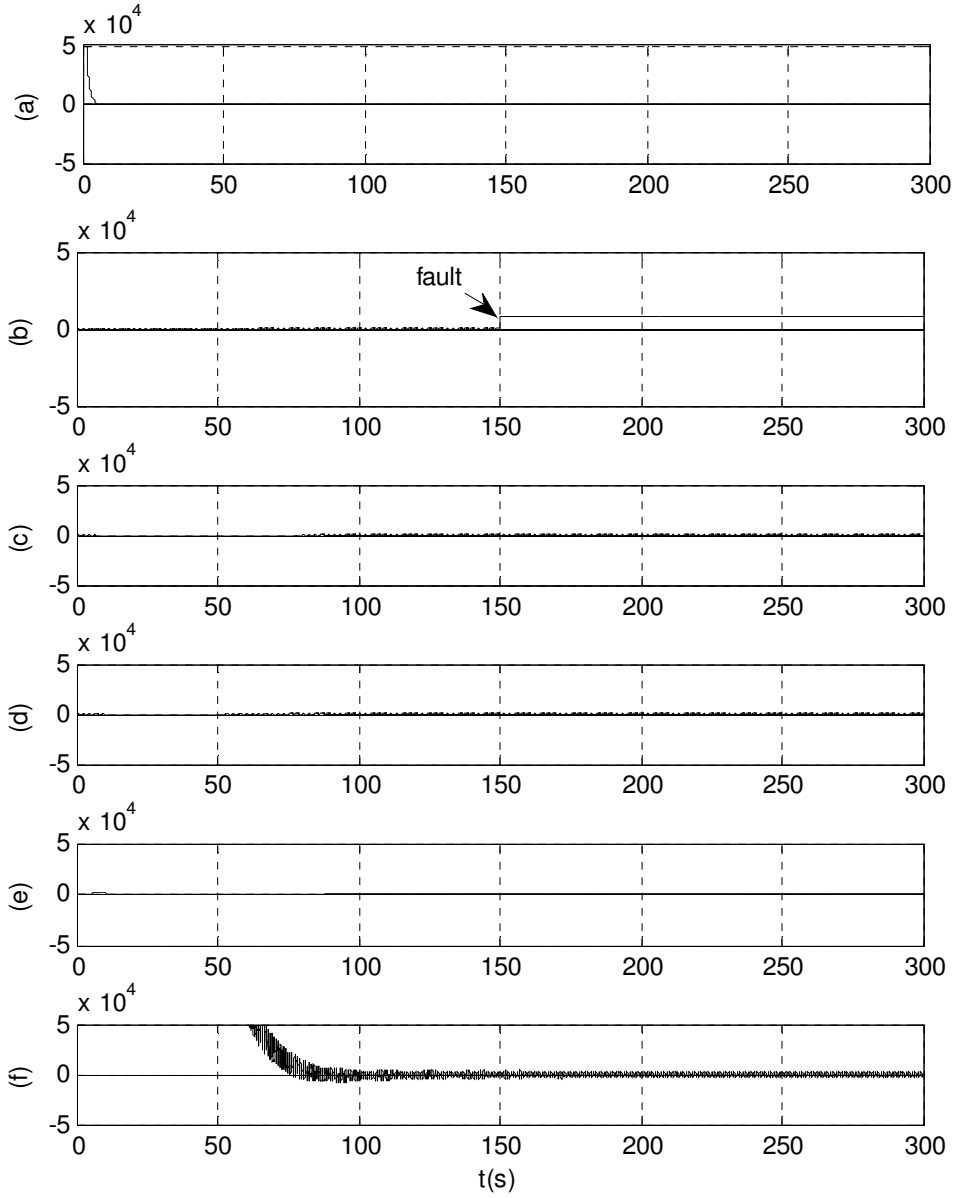


Figure 5.9: Residual norms when the difference between turbine rotor angle and generator rotor angle sensors is 8% at time-point t=150 s and with an effective wind speed of 10 m/s for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e), electric torque and (f) wind turbine torque

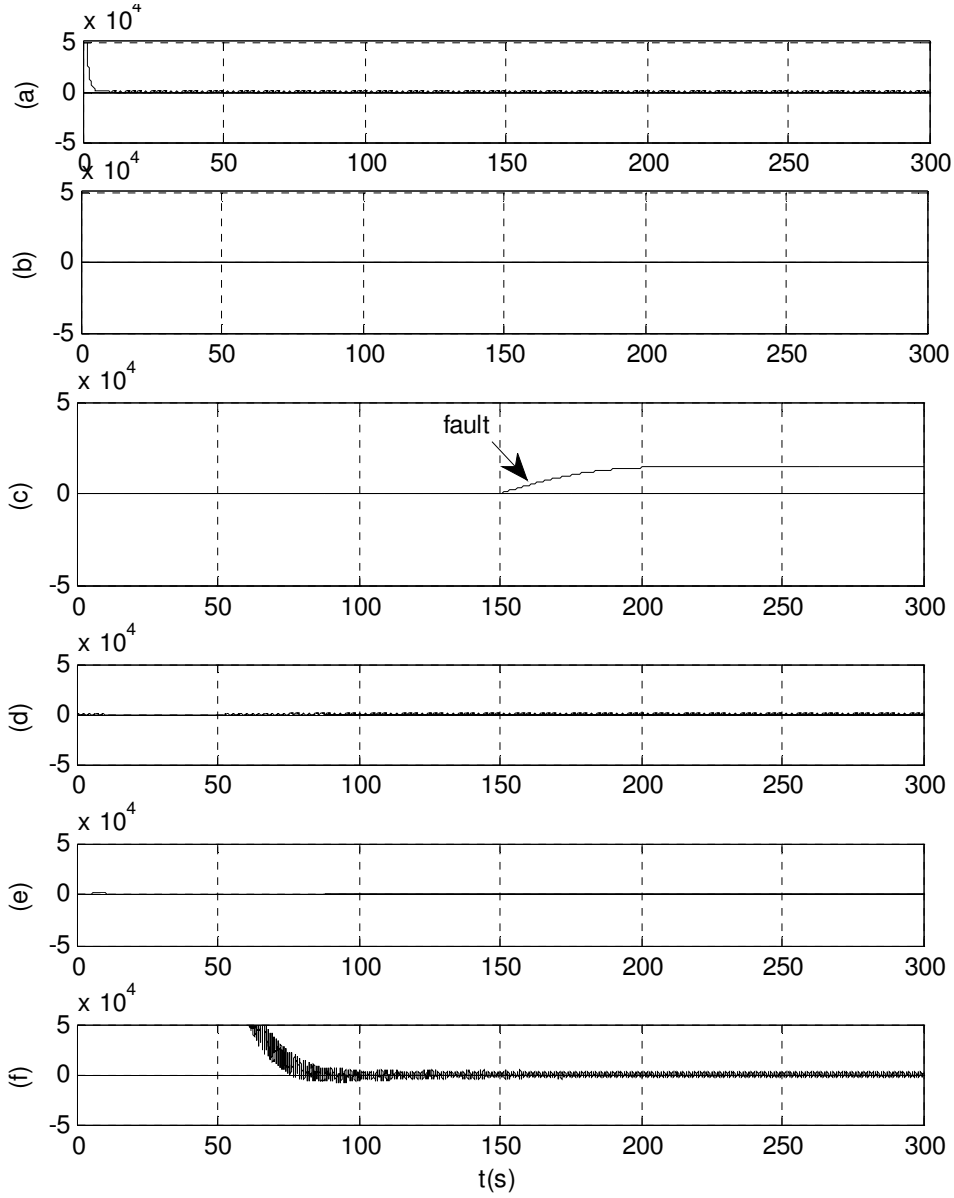


Figure 5.10: Residual norms when an 8% wind turbine rotor speed sensor occurs at time-point t=150 s with an effective wind speed of 10 m/s for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e), electric torque and (f) wind turbine torque

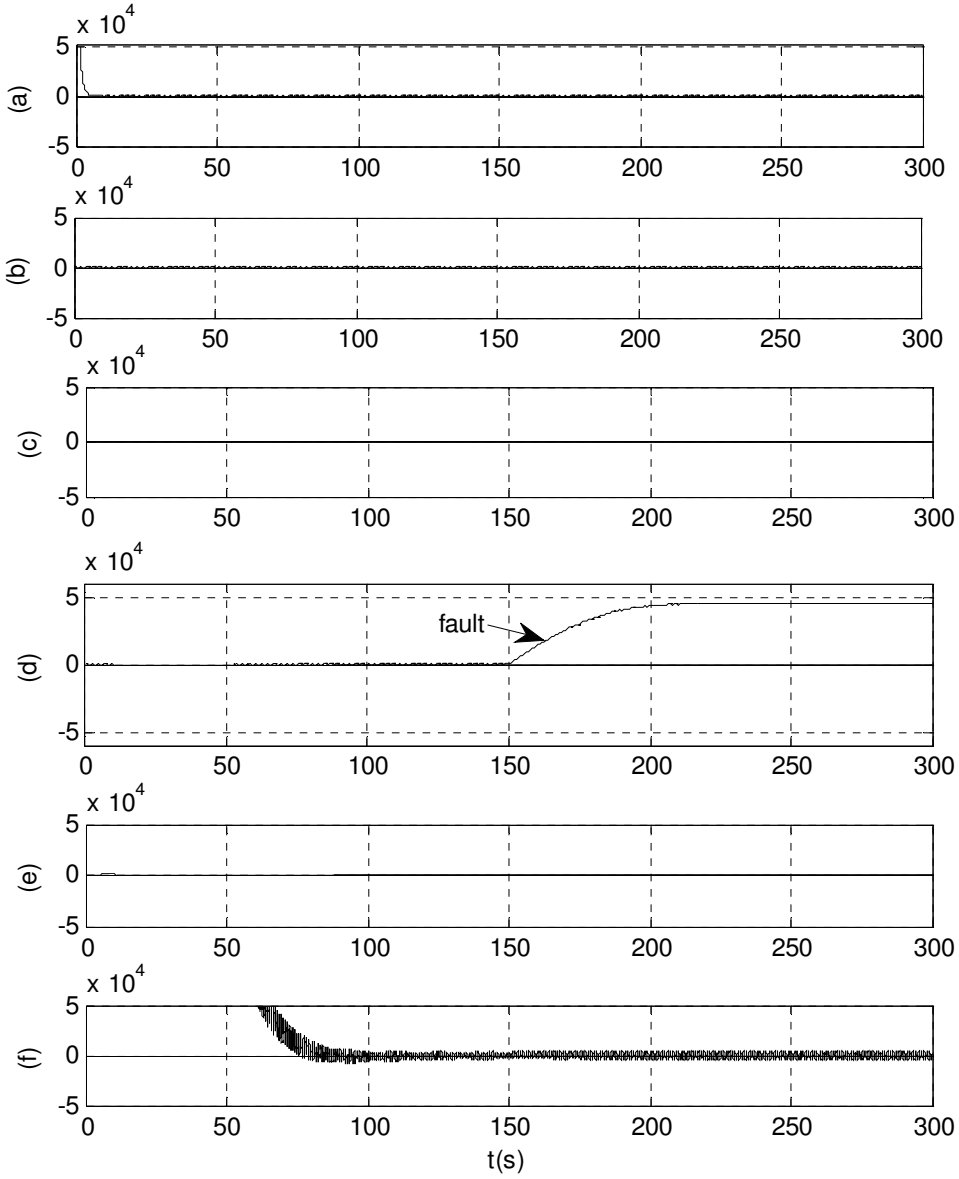


Figure 5.11: Residual norms when an 8% generator rotor speed sensor fault occurs at time-point t=150 s with an effective wind speed is 10m/s for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e), electric torque and (f) wind turbine torque

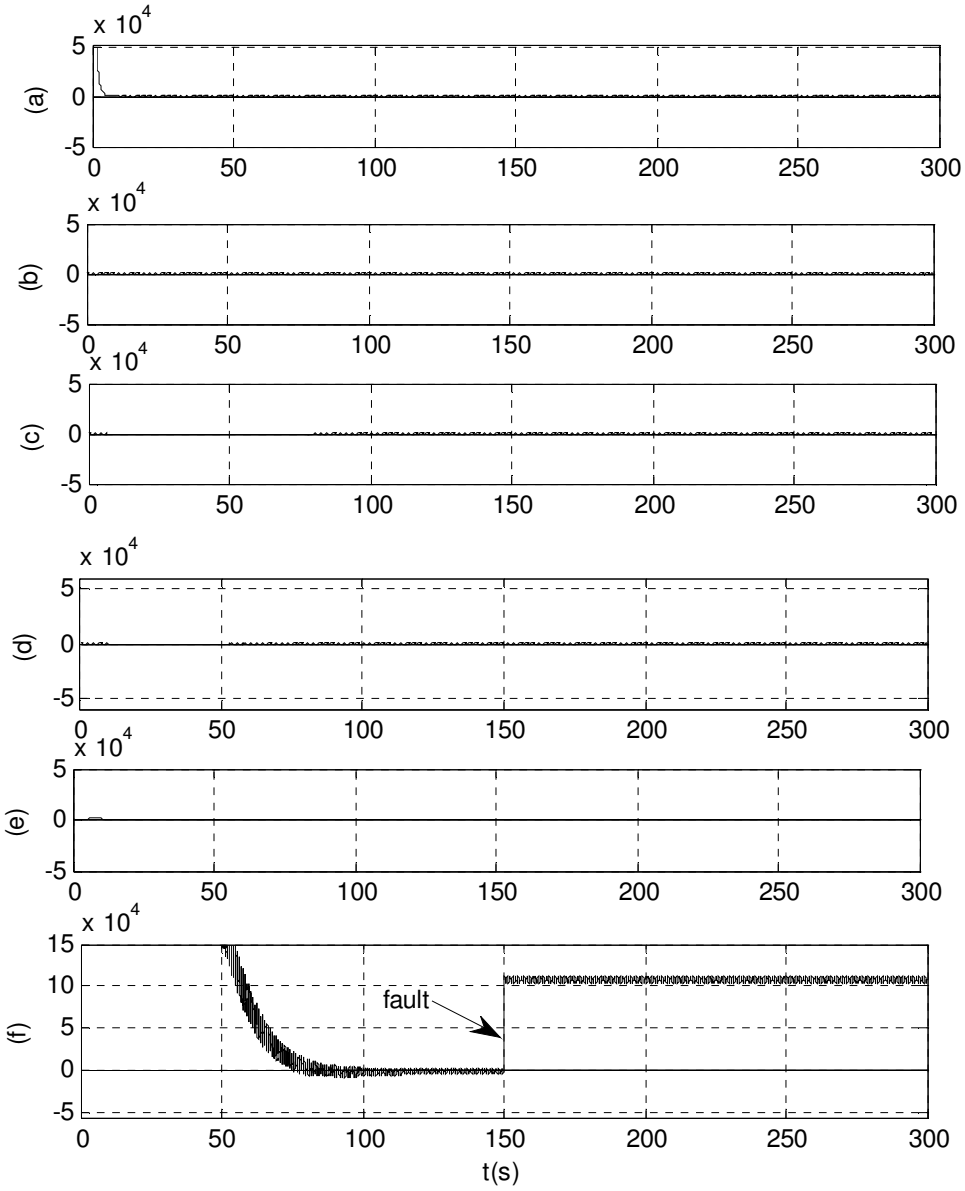


Figure 5.12: Residual norms when an 8% wind rotor torque sensor fault occurs at time-point t=150 s with an effective wind speed of 10 m/s for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e), electric torque and (f) wind turbine torque

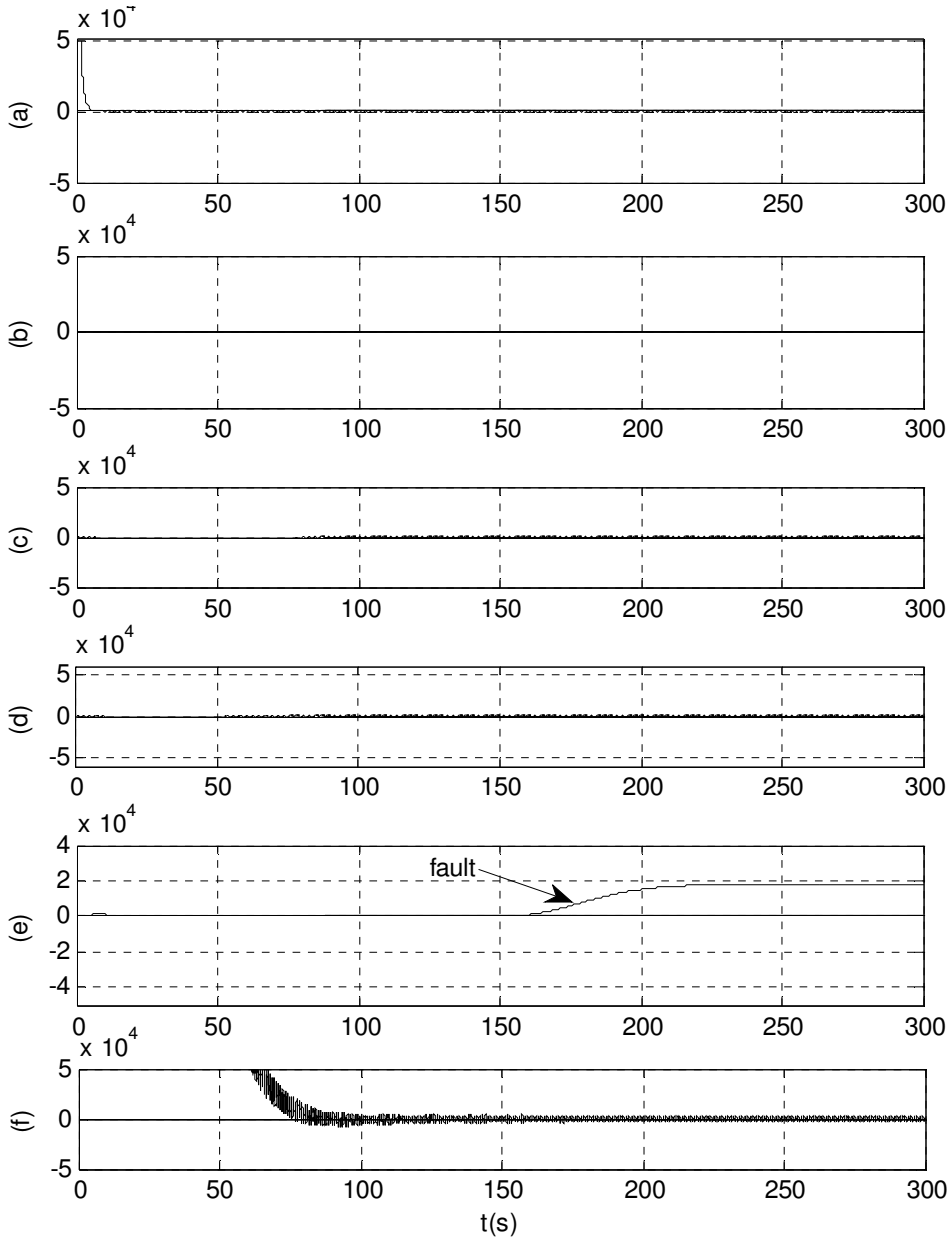


Figure 5. 13: Residual norms when an 8% electric torque sensor fault occurs at time-point t=150 s with an effective wind speed of 10 m/s for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e), electric torque and (f) wind turbine torque

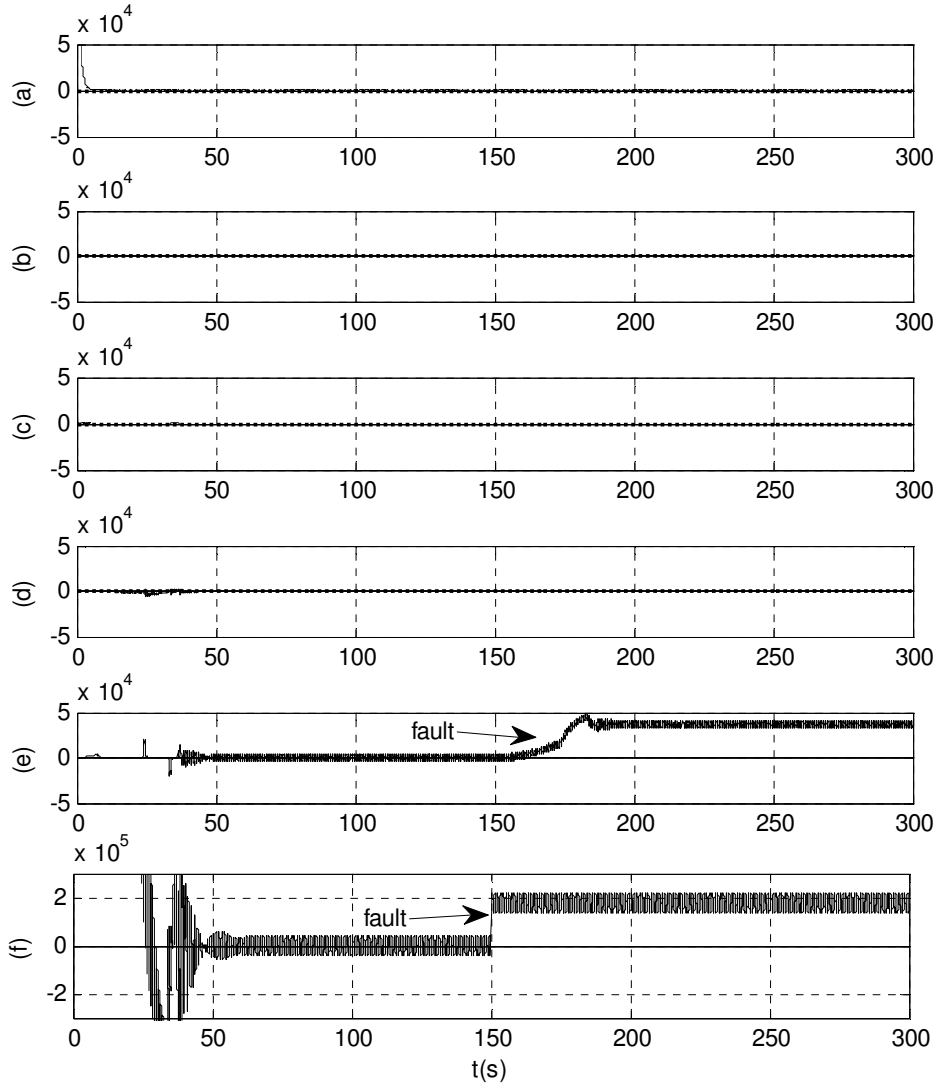


Figure 5.14: Residual norms when an 8% wind rotor torque and a 10% electric torque sensor fault occur at time-point t=150 s with an effective wind speed of 12 m/s for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e) electric torque and (f) wind turbine torque

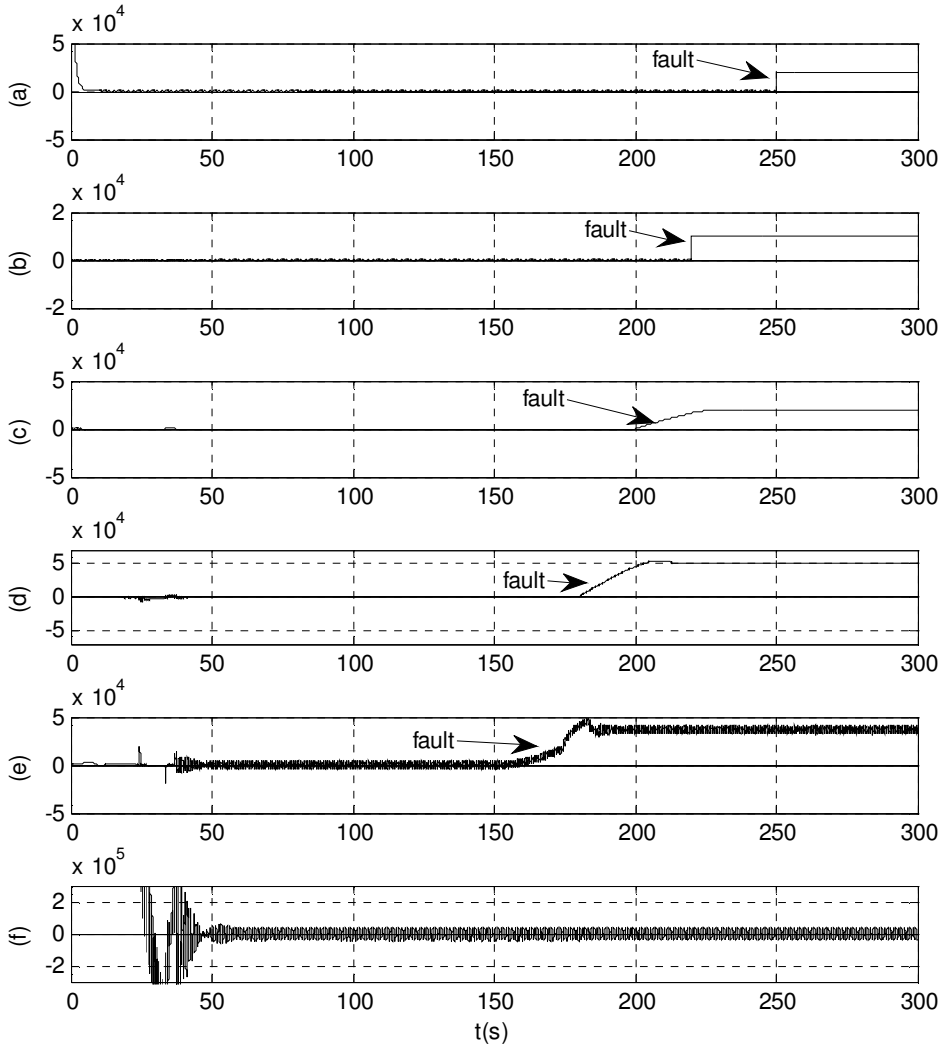


Figure 5.15: Residual norms for 10% pitch angle, 10% difference between turbine rotor angle and generator rotor angle, 10% wind turbine rotor speed, 9% generator rotor speed and 10% electrical torque sensor faults occurring at time-points t=250, 220, 200, 180 and 150 s, respectively, with an effective wind speed of 12 m/s for (a) pitch angle, (b) difference between turbine rotor angle and generator rotor angle, (c) wind turbine rotor speed, (d) generator rotor speed, (e) electric torque and (f) wind turbine torque

6 Conclusions and Future work

6.1Conclusions

The main objective of the work of this thesis was to design model-based fault detection and isolation techniques for large-scale wind turbines. The analytical model-based fault detection algorithms give an indication of faults by comparing measured outputs of the process to their estimations. The analytical model-based approach can be implemented in the same processor that implements the control algorithms; consequently, no additional hardware is required. Analytical model-based techniques use the knowledge of the monitored process and, therefore, are the most suitable approaches for fault detection. Such tecniques include observer-based approaches, which offer quick detection and require no excitation signal or online implementation.

In this thesis, model-based fault detection and isolation techniques for wind turbines were studied in detail, initially by an initial and extensive study to fault monitoring for linear systems; this proved to be extremely useful. If a process has strong non-linearities or the operating region is too wide, the linearisation error is too large to be handled by linear fault detection techniques. Therefore, a state-dependent non-linear observer-based scheme was developed for early FDI in variable-speed wind turbines.

The first part of the thesis focussed on developing a model of a 5MW wind turbine. This model was sufficiently detailed to be used as a simulation model (Chapter 2). The modelling procedure divided the overall wind turbine model into appropriate sub-models suitable of being modelled separately. These sub-models were:

- rotor torque,
- drive-train,
- doubly-fed induction generator,
- controller and
- fixed-point wind speed model.

Each wind turbine sub-model was presented and combined to obtain a completed non-linear wind turbine model. In order to control the non-linear model, two PI-based control design schemes were proposed for the blade-pitch angle controller in a variable-speed wind turbine (Chapter 3); an analytical method and a simulation-based method were developed to calculate the PI gains. The simulation results demonstrated good performance for both proposed PI schemes. The IMC-based PI design procedure was also applied for the control of current and electric torque. Figures B.15 and B.14 illustrate how subsystems were used to calculate the power coefficient, and optimal tip speed ratio allowed to design a rotor torque model without feedback of the rotor's rotational speed, making the proposed non-linear model a realistic model of a real wind turbine (the wind turbine subsystem converts wind speed into aerodynamic torque (see Figures B.13 and B14)).

The second part of the thesis developed linear and non-linear observer-based sensor fault detection and isolation methods for wind turbines. The fault detection and isolation method, which can be applied to a linear system, was optimised by using a MOGA to maximise the effect of faults and to minimise the effect of disturbances in the steady-state period on the residual (Chapter 4). The objectives were in conflict with each other, and were expressed in different units. Because of their nature, multi-objective optimization problems normally have not one but a set of solutions, known as Pareto points or Pareto optimal solutions (see Figures 4.17, 4.18 and 4.19).

To realise a complete linear observer-based sensor FDI, a Boolean decision table was constructed (see Table 4.1). If a fault occurs, the residual can be compared with this fault signature table and the location of the fault can be determined using a threshold. The selection of the threshold was very important. If the chosen threshold is too low, it results in false alarms, i.e. some disturbances will cause the residual to cross the threshold and result in an alarm. If the selected threshold is too high, small faults are undetected. Therefore, the threshold was selected slightly higher than the value of the evaluated residual signal in a fault-free case.

The rotor speed is variable in the case of below-rated wind speed and the behaviour of a wind turbine is non-linear so that it is necessary to apply non-linear fault detection techniques. A number of non-linear model-based fault diagnosis methods (see Table 5.1) were applied to monitoring wind turbines, including constructing the non-linearity from direct measurements and the use of an EKF and an UKF to detect faults in the wind turbine generator. In this thesis non-linear observer-based sensor fault detection and isolation methods were developed to monitor the wind turbine rotor, drive-train and generator. The state-dependent differential Riccati equation was used rather than the algebraic Riccati equations, which may pose an overly restrictive requirement on the observability and controllability of the system (Chapter 5). The fault detection system was optimised to be maximally sensitive to system faults and minimally sensitive to system disturbances and noise. The residual generator based on the non-linear observer was employed to develop a monitoring system, which was assessed using different additive sensor faults: pitch angle, the difference between turbine rotor angle and generator rotor angle, wind turbine rotor speed, generator rotor speed, electric torque and wind turbine torque faults. A dynamic threshold was designed for each sensor to identify the fault based on residual curves, with consideration given to the accuracy of each sensor; the output was one in the case of no sensor fault or zero in the case of a sensor fault. Simulation results demonstrated that this is a robust method for the detection and isolation of a fault in wind turbines. In addition, using the simulation results for fault detection in components and actuators (Chapter 4), these methods can be used to detect and locate the breakdown of components. In this case, where components are connected in the cascade, more than one fault may appear for a faulty component; it can be inferred that an actuator fault has occurred when the residual has the shape of an impulse signal.

The advantages of this scheme are that it is systematic to design and implement the algorithm in a practical way. Also, by using return of ‘1’ for each non-faulty sensor and ‘0’ for a faulty sensor, both single and multiple faults can be located quickly.

Compared to unknown input observers, this scheme is more relaxed. In this approach, instead of decoupling state estimations from unknown inputs, the residual

signal is made independent of unknown inputs such as the effective wind speeds on the blades of a wind turbine.

6.2 Future work

Future work concerning the non-linear observer-based fault diagnosis methods is proposed in Table 5.1. It is possible to apply the proposed methods to monitoring wind turbines, thus adding relevance for future research and practical applications.

The non-linear observer-based fault detection technique can be applied to monitor the vibrations of the tower and drive-train. Vibration monitoring is one of the most important aspects in wind turbine monitoring because it helps determine the condition of the rotating equipment. In a wind turbine, this equipment consists of the main bearing, the gearbox and the generator. In addition, vibration monitoring can be used on the turbine structure at the base and on the nacelle. This provides information concerning structural bending and the aerodynamic effect of the wind.

Finally, in monitoring wind turbines, it is recommended that the proposed non-linear observer-based fault detection and isolation method is used to support the CMS. The task is to predict when a machine fault occurs, diagnose or analyse data for better understanding and interpretation of fault detection, and to isolate a particular fault from others.

Bibliography

[1] Burton T., Sharpe D., Jenkin N., Bossanyi E. Wind Energy Handbook. Wiley, New York, 2001.

[2] Nichita C., Luca D., Dakyo B., Ceanga E. Large band simulation of the wind speed for real time wind turbine simulator. IEEE transactions on Energy Conversion, 2002, 17(4), 523-529.

[3] Vihriala H. Control of variable speed wind turbines. PhD Thesis. Tampere University of Technology, Finland, 2002.

[4] Bianchi F., De Battista H., Mantz R.J. Wind Turbine Control System – Principles, Modelling and Gain Scheduling Design. Spring, London, 2007.

[5] Leithead W.E., D.E. la Salle S., Reardon D. Role and objectives of control for wind turbines. IEE Proceedings –C, 138(2):135-148, 1991.

[6] Diop A., Ceanga E., Retiveau J., Methot J., Ilinca A. Real time three dimensional wind simulation for windmill rig tests. Renewable Energy, 32. 22682290, 2007.

[7] Wieringa, J. How far can agrometeorological station observations be considered representative? Preprint to 23rd Amer. Meteor. Soc. Conference on Agric. and Forest Meteor. (Albuquerque), 1998.

[8] Welfonder E., Neifer R., Spanner M., Development and experimental identification of dynamic models for wind turbine. Control Engineering Practice, 5(1):63-73, 1997.

[9] Munteanu I., Bratcu A., Cutulis N., Ceanga E. Optimal Control of Wind Energy System. Springer, London, 2008.

[10] Dumitrescu H., Georgescu A., Ceanga V., Popovici J., Ghiat Gh., Dumitrache A. Calculation of Propellers. Romanian Academy Publishing House, Bucharest, 1990.

[11] Le Gourieres D. Theory, design and practical calculation of wind energy systems. Wind Energy, Paris, 1982.

[12] Rodriguez J.L., Rodriguez F., Burgos J.C., Chincilla M., Arnalte S. Experimental rig to emulate wind turbines. In Proceeding of the ICEM Conference, 3:2033-2038, 1998.

[13] Molenaar D-P. Cost-effective design and operation of viable speed wind turbines. PhD Thesis. Technical University of Delft, Netherlands, 2003.

[14] Wilkie J., Leithead W.E., Anderson C. Modelling of wind turbines by simple models. Wind Engineering, 4:247-274, 1990.

[15] Heier S. Wind Energy Conversion Systems. John Wiley and Sons Ltd, Chichester, England, 1998.

[16] Wang C. Control, stability analysis and grid integration of wind turbines. PhD Thesis. Imperial College London, 2008.

[17] Johnson K.E., Pao L.Y., Balas M.J., Fingersh L.J. Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture. IEEE Control Systems Magazine, 26(3):70–81, 2006.

[18] Pena R., Clare J.C., Asher G.M. Doubly-fed induction generator using back-to-back pwm converters and its application to variable-speed wind-energy generation. IEE Proc. Electr. Power Appl., 143(3):231–241, 1996.

[19] Bossanyi E.A. Wind turbine control for load reduction. Wind Energy, 6:229–244, 2003..

[20] Wright A.D., Balas M.J. Design of controls to attenuate loads in the controls advanced research turbine. Journal of Solar Energy Engineering, 126:1083–1091, 2004.

[21] Krough T. HAWC load simulation of generic 5MW offshore wind turbine model. Riso National Laboratory, Roskilde, 2004.

[22] Suryanarayanan S., Dixit A. A procedure for the development of control oriented linear models for horizontal-axis large wind turbines. Journal of Dynamic Systems, Measurement and Control, ASME, 129:469–479, 2007.

[23] Hansen M.H., Hansen A., Larsen T.J., Seoye S., Seoensen P., Fuglsang P. Control design for a pitch-regulated, variable speed wind turbine. Technical report. Riseo National Laboratory, Roskilde, 2005.

[24] Hansen A.D., Soerensen P., Blaabjerg F., Becho J. Dynamic modelling of wind farm grid interaction. Wind Engineering, 26(4):191–208, 2002.

[25] Bianchi F.D., Mantz R.J., Christiansen C.F. Gain scheduling control of variablespeed wind energy conversion systems using quasi-LPV models. Control Engineering Practice, 13(2):247–255, 2005.

[26] Rasila M. Torque and speed control of a pitch regulated wind turbine. Master's Dissertation. Department of Electric Power Engineering, Chalmers University of Technology, Sweden, 2003.

[27] Muljadi E., Butterfield C.P. Pitch-controlled variable-speed wind turbine generation. IEEE Trans. Industry Applications, 37(1):240–246, 2001.

[28] Lubosny Z. Wind Turbine Operation in Electric Power Systems. Springer-Verlag, Berlin, 2003.

[29] Blaabjerg F., Chen Z., Kjaer S.B. Power electronics as efficient interface in dispersed power generation systems. IEEE Trans. Power Electronics, 19(5):1184, 119, 2004.

[30] Akhmatov V. Variable-speed wind turbines with doubly-fed induction generators. Part i: Modelling in dynamic simulation tools. Wind Energy, 26(2): 85 – 108, 2002.

[31] Morren J., de Haan S.W.H., Bauer P., Pierik J.T.G. Comparison of complete and reduced models of a wind turbine with doubly-fed induction generator. In Proceeding of the 10th European conference on Power Electronics and applications (EPE), Toulouse, France, 2003.

[32] Machmoum M., Poitiers F., Darengosse C., Queric A. Dynamic performances of a doubly-fed induction machine for a variable-speed wind energy generation. International Conference on Power System Technology, 2002. Proceedings, PowerCon 2435-2431, 2002.

[33] Vas P. Sensorless Vector and Direct Torque Control. Oxford University Press, 1998.

[34] Ackerman T. Wind Power in Power System. John Wiley, Chicester, UK, 2005.

[35] Bailey, B.H. New York State Wind Energy Handbook. New York State Wind Energy Office, 1982.

[36] Zhao Y., Zou X.D., Xu Y.N., Kang,Y., Chen J. Maximal power point tracking under speed-mode control for wind energy generation system with doubly-fed introduction generator. Power Electronics and Motion Control Conference, 2006. IPEMC 2006. CES/IEEE 5th International, 1, 1-5, 2006.

[37] Giebhardt J., Hahn B., Durstewitz M. Requirements for condition based operation and maintenance in offshore wind farms. ISET, BOW, Berlin, 2007.

[38] Boukhezzara B., Lupua L., Siguerdidjanea, H., Hand M., Multivariable control strategy for variable speed, variable pitch wind turbines. Renewable Energy, 32:1273–1287, 2007.

[39] Eisenhut C., Krug F., Schram C. Wind-turbine model for system simulations near cut-in wind speed". IEEE Transactions on Energy Conversion, 2007, 22(2), 414-420.

[40] Zhang J., Cheng M., Chen Z., Fu X. Pitch angle control for variable speed wind turbines. DRPT 2008, China, 2008.

[41] Jonkman J., Butterfield S., Musial W., Scott G. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060. February 2009.

[42] Hongwei L., Yonggang L., Wei, L. Study on control strategy of individual blade pitch-controlled wind turbine. Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, 2006, vol. 2, pp. 6489-6492.

[43] Holdsworth L., Wu X.G., Ekanayake J.B. Jenkins N. Comparison of fixed speed and doubly-fed induction wind turbines during power system disturbances. Proc. Inst. Elect. Eng., Commun., 150(3):343–352, 2003.

[44] Hudson R.M., Stadler F., Seehuber M. Latest developments in power electronic converters for megawatt class wind turbines employing doubly-fed generators, In Proc. Int. Conf. Power Conversion, Intelligent Motion, Nuremberg, Germany, 2003, 173-179.

[45] Ribrant J., Bertling L. Survey of failures in wind power systems with focus on Swedish wind power plants during 1997-2005. Power Engineering Society General Meeting, 2007. IEEE , 1-8, 2007

[46] Patton R., Chen J. A review of parity space approaches to fault diagnosis. In Proc. IFAC/IMACS Sympo. SAFEPROCESS'91", 1:239-255, 1991.

[47] Venkatasubramanian V., Rengaswamy R., Yin K., Kavuri S.N. A review of process fault detection and diagnosis part I: quantitative model based methods; Computers and Chem. Eng. 27:293-311, 2003.

[48] Venkatasubramanian V., Rengaswamy R., Yin K., Kavuri S.N. A review of process fault detection and diagnosis part II: qualitative models and search strategies; Computers and Chem. Eng. 27:313-326, 2003.

[49] Venkatasubramanian V., Rengaswamy R, Yin K., Kavuri S.N. A review of process fault detection and diagnosis part III: process history based methods; Computers and Chem. Eng. 27:327-346, 2003.

[50] Lucente M. Condition monitoring system in wind turbine gearbox, Master's Dissertation, KTH, Royal Institute of Technology, 2008.

[51] Milne R. Strategies for diagnosis; IEEE Trans. on Syst., Man and Cyber. 17(3):333-339, 1987.

[52] Isermann R., Ballé P. Trends in the application of model-based fault detection and diagnosis of technical processes; Control Eng. Practice, 5(5):709–719, 1997.

[53] Ducard G.J.J. Fault-tolerant flight control and guidance systems: practical methods for small unmanned aerial vehicles; Advances in Industrial Control, Springer, 2009.

[54] Blanke M., Kinnaert M., Lunze J., Staroswiecki M. Diagnosis and Fault-Tolerant Control; Springer, 2006.

[55] Frank P.M. Analytical and qualitative model-based fault diagnosis - a survey and some new results; Eur. J. Control, 2:6–28, 1996.

[56] Frank P.M. On-line fault detection in uncertain non-linear systems using diagnostic observers: a survey. Int. J. Systems Sci., 25(12):2129–2154, 1994.

[57] Frank P.M. Advanced fault detection and isolation schemes using non-linear and robust observers; In "10th IFAC World Congress", Munich, Germany, 1987, 63-68.

[58] Frank P.M. Enhancement of robustness in observer-based fault detection; In "Proc. IFAC/IMACS Sympo. SAFEPROCESS'91", 1:275-287, 1991.

[59] Beard R.V. Failure accommodation in linear system through self reorganization. PhD thesis, Massachusetts Institute of Technology, 1971.

[60] Clark R.N. The dedicated observer approach to instrument failure detection; In Proc. of the 18th IEEE Conf. on Decision & Contr., 237-241, Fort Lauderdale, FL, USA, 1979.

[61] Patton R.J., Kangethe S.M. Fault Diagnosis in Dynamic Systems, Theory and Application, Chapter 4:99-154; Prentice Hall, 1989.

[62] Patton R.J., Chen J. Observer-based fault detection and isolation: Robustness and applications; Contr. Eng. Practice, 5(5):671-682, 1997.

[63] Mehra R.K., Peschan J. An innovation approach to fault detection and diagnosis systems; Automatica, 7:637-640, 1971.

[64] Willsky A.S. A survey of design methods for failure detection in dynamic systems; Automatica, 12(6):601- 611, 1976.

[65] Zolghadri A. An algorithm for real-time failure-detection in Kalman filters; IEEE Trans. on Automat. Contr., 41(10):1537-1539, 1996.

[66] Said M.S.N., Benbouzid M.E.H., Benchaib A. Detection of broken bars in induction motors using an extended Kalman filter for rotor resistance sensor less estimation; IEEE Trans. on Energy Conversion, 15(1):66-70, 2000.

[67] O'Reilly J. Observers for Linear Systems. Academic Press, 1983)

[68] Seliger R., Frank P.M. Fault diagnosis by disturbance decoupled non-linear observers. In Proc. 30th IEEE Conf. on Decision & Contr., 3:2248-2253, Brighton, England, 1991.

[69] Seliger R., Frank P.M. Robust residual evaluation by threshold selection and a performance index for non-linear observer-based fault diagnosis. In Int. Conf. on Fault Diagnosis (Tooldiag'93), Toulouse, France, 1993, 496-504.

[70] Bastin G., Gevers M.R. Stable adaptive observers for non-linear time varying systems. IEEE Trans. on Automat. Contr., 650-658, 1988.

[71] Hamelin F., Sauter D. Robust residual generation for FDI in uncertain dynamic systems. In Proc. 34th IEEE Conf. on Decision & Contr, New Orleans, USA, 1995.

[72] Frank P.M., Keller L. Sensitivity discrimination observer design for instrument failure detection. IEEE Trans. Aero. Electron. Syst., 16(4):460–467, 1980.

[73] Gertler J., Singer D. A new structural framework for parity equation based failure detection and isolation. Automatica, 26(2), 381.388, 1990.

[74] Gertler J., Luo Q., Anderson K., Fang X.W. Diagnosis of plant failures using orthogonal parity equations. In Proc. of the 11th IFAC World Congress, Tallin, 1990.

[75] Gertler J., Monajemy R. Generating directional residuals with dynamic parity relations. Automatica, 31(4):627.635, 1995.

[76] Chow E.Y., Willsky A.S. Analytical redundancy and the design of robust failure detection systems. IEEE Trans. on Automat; Contr. (7),603. 614, 1984.

[77] Isermann R. Process fault detection on modelling and estimation methods - a survey. Automatica, 20(4):387–404, 1984.

[78] Jiang T., Khorasani K., Tafazoli S. Parameter estimation-based fault detection, isolation and recovery for non-linear satellite models. IEEE Trans. Control Syst. Technol., 16(4):799–808, 2008.

[79] Isermann R. Fault-diagnosis systems: an introduction form fault detection to fault tolerance. Springer, 2006.

[80] Garcia E.A., Frank P.M. On the relationship between observer and parameter identification based approaches to fault detection. In Proc. 13th IFAC World Congress, pages 25–29, San Francisco, USA, 1996.

[81] Ding S.X. Model-based fault diagnosis techniques - design schemes, algorithms and tools. Springer, 2008.

[82] Khan A.Q., Ding S.X. Threshold computation for robust fault detection in a class of continuous-time non-linear systems. In Proc; European Contr. Conf., 3088–3093, Budapest, Hungary, 2009.

[83] Seliger R., Frank P.M. Robust residual evaluation by threshold selection and a performance index for non-linear observer-based fault diagnosis. In International Conference on Fault diagnosis, 496–504, Toulouse, France, 1993.

[84] Seliger R., Frank P.M. Robust observer-based fault diagnosis in non-linear uncertain systems; In R. J. Patton, P. M. Frank, and R. N. Clark, editors, Issues of fault diagnosis for dynamic systems, 145–187; Springer, 2000.

[85] Ogata K. Modern Control Engineering. Pearson Education International, fourth edition, 2002.

[86] Gertler J. Analytical redundancy methods in fault detection and isolation; survey and synthesis. IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden, Germany, 9–21, 1991.

[87] Chiang, L H., Braatz, R. D., Russell, E. L. et al. Fault Detection and Diagnosis in Industrial Systems. Advanced Textbooks in Control and Signal Processing, Springer, 2001.

[88] Hur S., Katebi M.R., Taylor A. Fault detection and diagnosis of a plastic film extrusion process. In: UKACC International Conference on CONTROL, Coventry, UK, 2010, pp. 1-6.

[89] Chankong V., Haimes Y. Multi-objective decision making theory and methodology. New York: North-Holland, 1983.

[90] Burrows S.P., Patton R.J. Design of a low sensitivity, minimum norm and structurally constrained control law using eigenstructure assignment. Optimal Control Appl. and Meth., 12:131-140, 1991.

[91] Chen J., Patton R.J. Robust model-based fault diagnosis for dynamic systems. Kluwer Academic Publishers, 1999.

[92] Hengy D., Frank P.M. Component failure detection using local second order observers. 2nd European Workshop on Fault Diagnostics, Reliability and Related Knowledge-based Approaches, Manchester, UK, 1987.

[93] Frank P.M. Advanced fault detection and isolation schemes using non-linear and robust observers, l0th IFAC World Congress, Munich, 1987, 63-68.

[94] Frank P.M., Ding S.X. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. J. Process Control, 7:403–424, 1997.

[95] Witczak M. Modelling and estimation strategies for fault diagnosis of non-linear systems: from analytical to soft computing approaches. Lecture Notes in Control and Information Sciences. Springer, 2007.

[96] Garcia E.A., Frank P.M. Deterministic non-linear observer-based approaches to fault diagnosis: a survey. Control Eng. Practice, 5(5):663–670, 1997.

[97] Frank P.M. Enhancement of robustness in observer-based fault detection. Int. J. Control, 59(4):955–981, 1994.

[98] Frank P.M., Schreier G., Garcia E.A. New Directions in Non-Linear Observer Design, volume 244/1999 of Lect. Notes Contr. Inform. Sci., chapter Non-linear observers for fault detection and isolation, pages 399–422. Springer Berlin / Heidelberg, 1999.

[99] Frank P.M. Advances in observer-based fault diagnosis in dynamic systems. Engineering Simulation, 13:717–760, 1996.

[100] Heng D. Frank P.M. Component failure detection via non-linear observers. In Proc. IFAC Workshop on Fault detection and safety of chemical plants, 153–157, Kyoto, Japan, 1986.

[101] Frank P.M. Fault diagnosis in dynamic systems using analytical and knowledge based redundancy-a survey and some new results. Automatica, 26:459–474, 1990.

[102] Adjallah K., Maquin D.J. Ragot. Non-linear observer-based fault detection. In Proceedings of the Third IEEE Conference on Control Applications, 1115–1120, Glasgow, UK, 1994.

[103] Tsinias J. Observer design for non-linear systems. Systems and Control Letters, 13:135-142, 1989.

[104] Thau F.E. Observing the state of non-linear dynamic systems, Internat. J. Control 17, 1973, 471-479

[105] Heraud N., Kahyeh M.A., Guelle I.S. Monitoring of the DFIG of a variable speed wind turbine. In Transmission and Distribution Conference and Exposition: Latin America (T&D-LA), 2010 IEEE/PES, 2010, pp. 440 – 445.

[106] Willems J.C. Stability theory of dynamical systems. Wiley, New York, 1970.

[107] Zeitz M. The extended Luenberger observer for non-linear systems. Syst. Control Lett., 9(2):49-156, 1987

[108] Friedland B. Advanced control system design, Prentice-Hall, Upper Saddle River, NJ, 1996.

[109] Williams D., Friedland B., Madiwale A.N. Modern control theory for design of autopilots for bank-to-turn missiles, AIAA Journal of Guidance, Control, and Dynamics, 10(4):378-386, 1987.

[110] Cloutier J.R., D'Souza C.N., Mracek C.P. "Non-linear regulation and non-linear Hinf control via the state-dependent Riccati equation technique: part 2, Examples", First Int. Conf on Non-linear Problems in Aviation and Aerospace, Daytona Beach, FL, 1996.

[111] Haessig D.A., Friedland B. State-dependent differential Riccati equation for non-linear estimation and control. In Proc. 15th Triennial World Cong., Barcelona, Spain, IFAC, 2002.

[112] Bennouna N., Heraud H., Camblong M., Rodriguez M., Kahyeh M.A. Diagnosis and fault signature analysis of a wind turbine at a variable speed. Journal of Risk and Reliability, 223:41-50, 2009.

[113] Glumineau C.H., Moog F., Plestan, New algebra-geometric conditions for the linearization by input-output injection, IEEE Trans. Automat. Control, 41:598–603, 1996.

[114] Krener A.J., Isidori A. Linearization by output injection and non-linear observers. Systems & Control letters, 3:47-52, 1983.

[115] Krener A.J., Respondek W. Non-linear observers with linearizable error dynamics. SIAM Journal on Control and Optimization, 23(2):197-216, 1985.

[116] Azad S.P., Tate J.E. Parameter estimation of doubly-fed induction generator driven by wind turbine. In Power Systems Conference and Exposition (PSCE), IEEE/PES, 2011.

[117] Kim Y.-R., Sul S.-K., Park, M.-H. Speed sensorless vector control of induction motor using extended Kalman filter. IEEE Trans. Industry Applications, 30(5) :225–1233, 1994.

[118] Barut M., Bogosyan S., Gokasan M. Speed-sensorless estimation for induction motors using extended Kalman filters. IEEE Trans. Industrial Electronics, 54(1):272–280, 2007.

[119] Bishop G., Welch G. An introduction to the Kalman filter. SIGGRAPH Journal, 2001.

[120] Yang Q. Model-based and data-driven fault diagnosis methods with applications to process monitoring. PhD Thesis. Case Western Reserve University, 2004.

[121] Julier S., Uhlmann J. A new extension of the Kalman filter to non-linear systems. In Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, 1997.

[122] Wan E., Van Der Merwe R. The unscented Kalman filter for non-linear estimation. in Proc. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC), 153–158, 2000.

[123] Arulampalam M.S., Maskell S., Gordon N., Clapp T. A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing, 50(2):174-188, 2002.

[124] Gordon N.J., Salmond D.J., Smith A.F.M. Novel approach to non-linear/non-Gaussian Bayesian state estimation. Radar and Signal Processing, IEE Proceedings F, 140(2):107-113, 1993.

[125] Kitagawa G. Monte Carlo filter and smoother for non-Gaussian non-linear state space models. Journal of Computational and Graphical Statistics, 5(1):1–25, 1996.

[126] Krokavec D., Filasova A. On observer-based residual generator design for a class of non-linear systems. In IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 2011.

[127] Clark R.N., Fosth D.C., Walton V.M. Detecting instrument malfunctions in control systems. IEEE Transactions on Aerospace and Electronic Systems, 11(4): 465-473, 1975.

[128] Kadirkamanathan, V., Li, P., Jaward, M. H., Fabri, S. G. A sequential Monte Carlo filtering approach to fault detection and isolation in non-linear systems. In Proceedings of the 39th IEEE Conference on Decision and Control, 2000.

[129] Anderson B.D.O., Moore J.B. Optimal filtering. Prentice-Hall, Englewood Cliffs, NJ, 1979.

[130] Kitagawa G.A. Self-organizing state space model. Journal of the American Statistical Association, 93(443):1203- 1215, 1998.

[131] Zarei J., Poshtan J. Design of non-linear unknown input observer for process fault detection. Industrial & Engineering Chemistry Research, 49(22):11443-11452, 2010.

[132] Dormand J.R., Prince P.J. "A family of embedded Runge-Kutta formulae," J. Comp. Appl. Math., 6:19-26, 1980.

[133] Zakian V., Al-Naib U. Design of dynamical and control systems by the method of inequalities. Proc. IEE-D, 120:1421-1427, 1973.

[134] Gertler J. Fault detection and isolation using parity relations. Contr. Eng. Practice 5(5):653-661, 1997.

Appendix A: Parameters of a 5MW wind turbine

Table A. 1: Nominal physical parameters of a 5MW wind turbine [16]

Description	Parameter	Value
Rated turbine power	P_N	5MW
Number of blades	N_B	3
Turbine blade length	R	55m
Turbine inertia	J_T	2:225* 10^7 kgm^2
Gearbox ratio	n_g	60.88
Generator inertia	J_G	600 kgm^2
Torsional stiffness	K_s	7.5 *10^8 Nm/rad
Torsional damping	C_s	100 Nms/rad
Damping coefficient	b_d	0 kgm^2/s
Air density	ρ	1.225 kg/m^3
Grid frequency	f	50Hz
Synchronous speed	ω_s	104.7 rad/s
Stator resistance	R_s	0.0022 Ω
Rotor resistance	R_r	0.0018 Ω
Stator inductance	L_s	3 H
Rotor inductance	L_r	2.9 H
Mutual inductance	L_m	2.9 H
Pole pairs	n_p	3
Stator voltage	V_s	690V (RMS)
Hub height	Z	125 m
Roughness length	z_0	0.0002

Appendix B: Simulink models

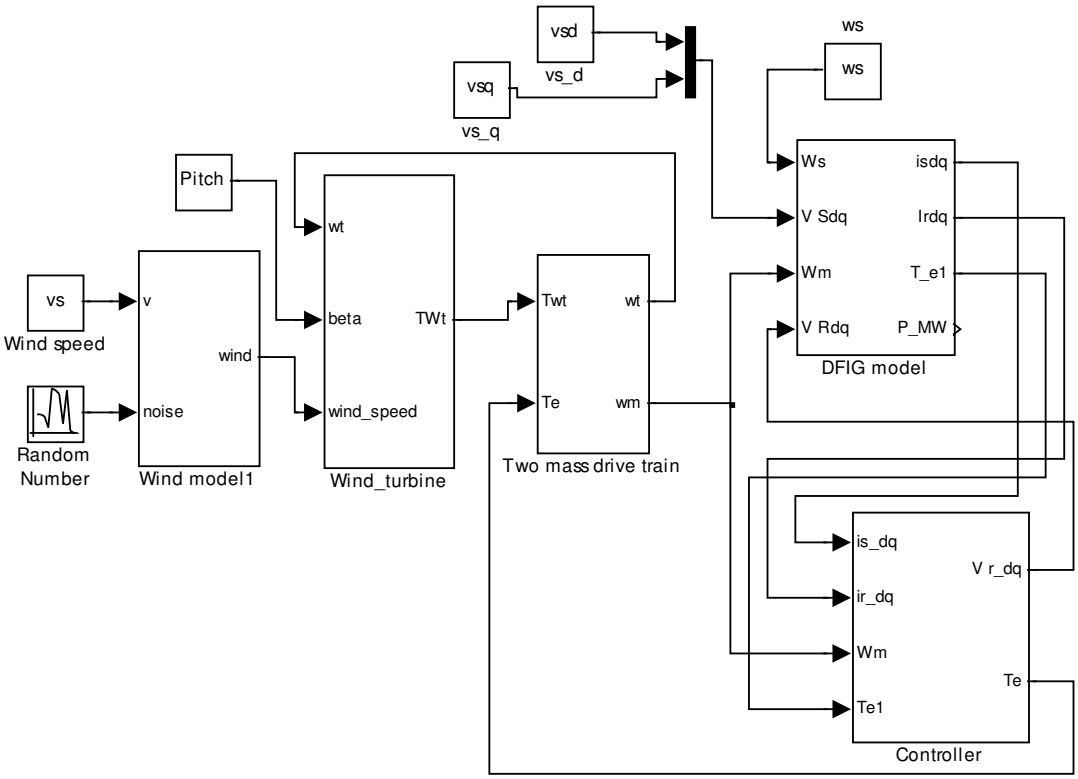


Figure B. 1: Simulink model for all subsystems of the wind turbine

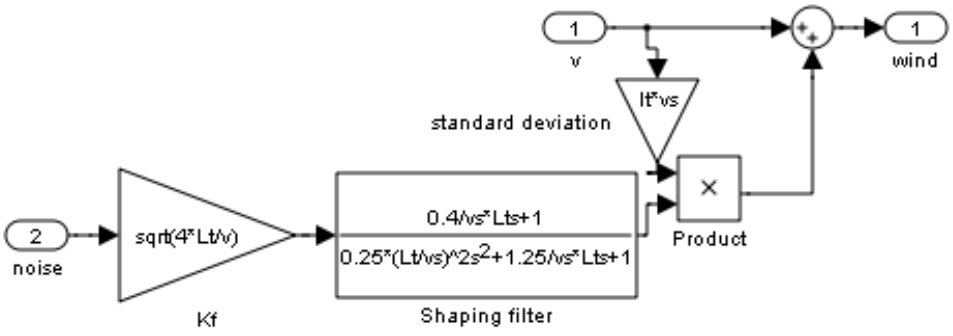


Figure B. 2: Simulink block diagram of the non stationary wind speed

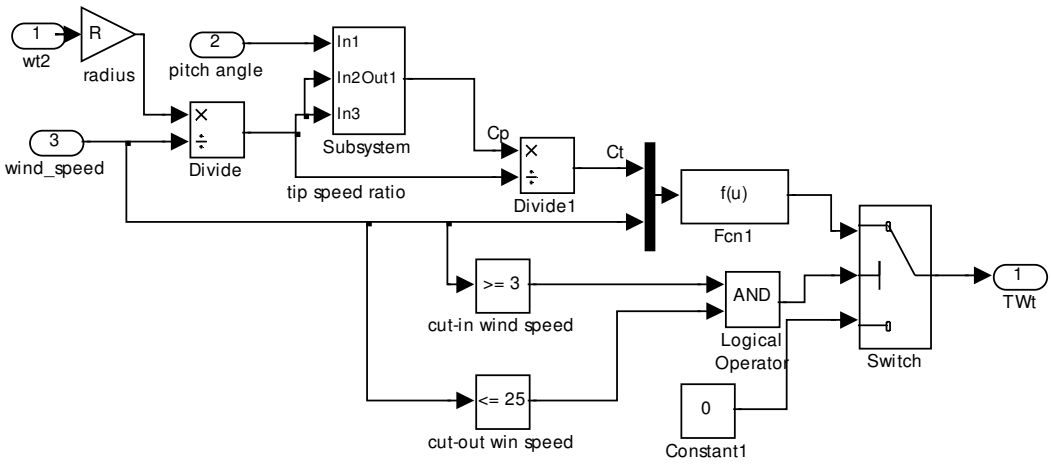


Figure B. 3: Simulink block diagram model of the wind turbine rotor

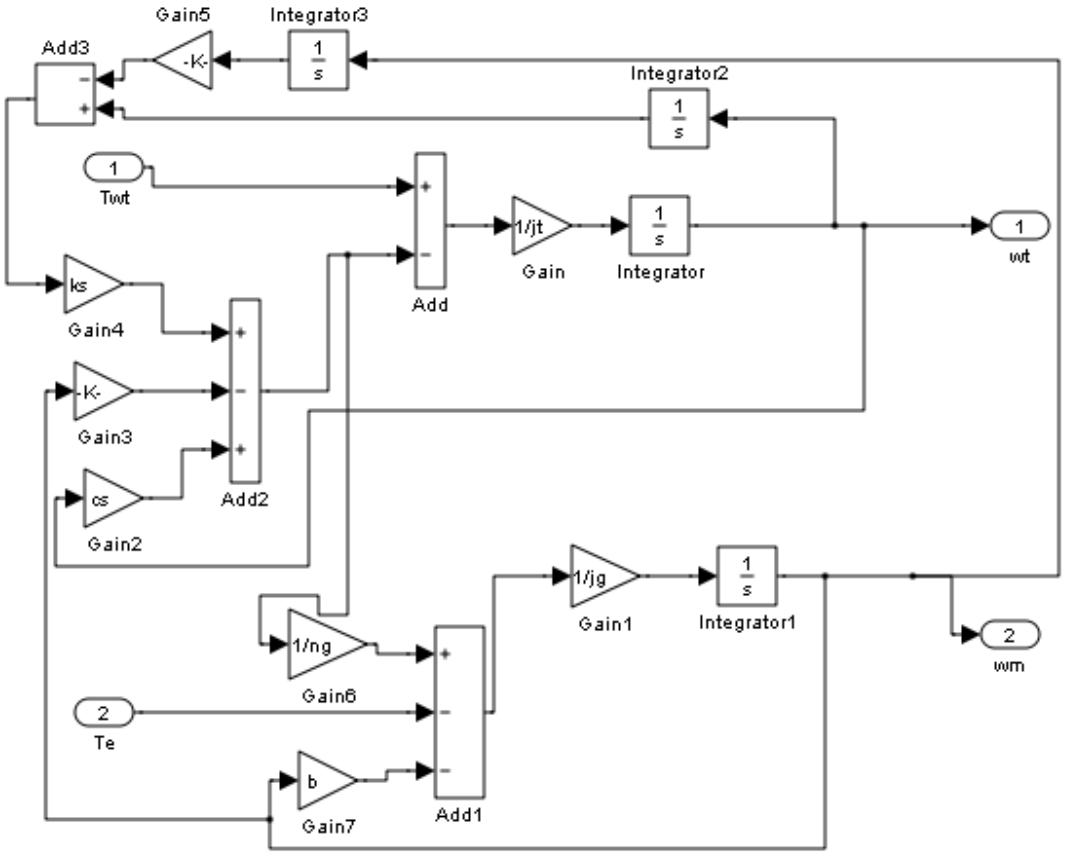


Figure B. 4: Simulink model of the two mass drive-trains

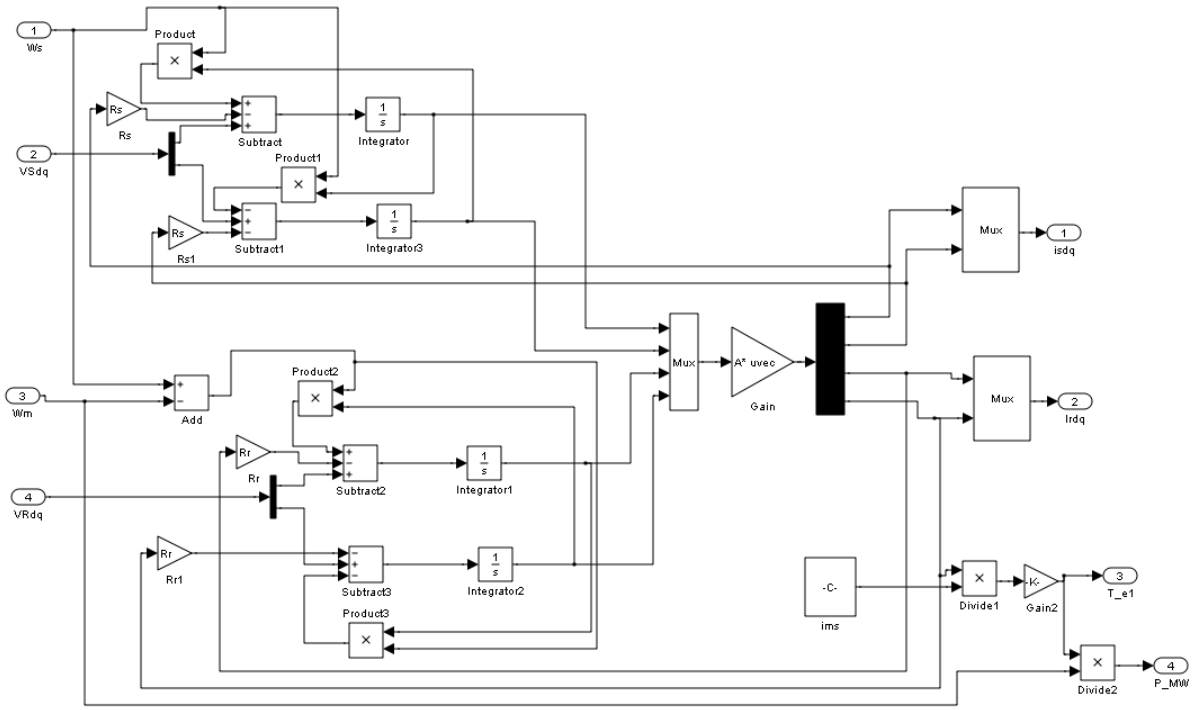


Figure B. 5: Simulink model of the 4*th* order DFIG

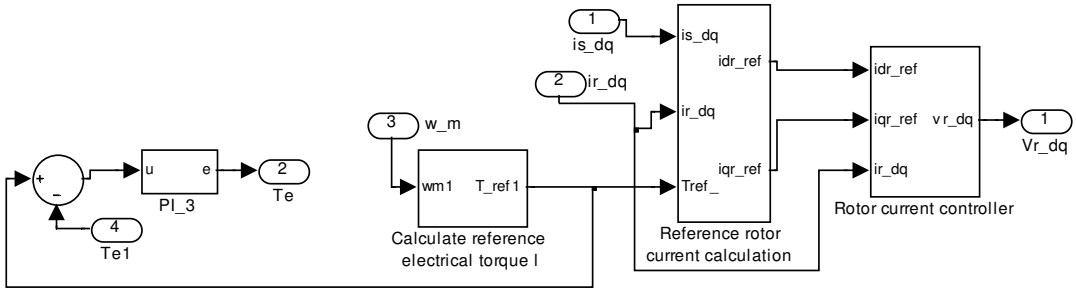


Figure B. 6: Simulink model of the controller

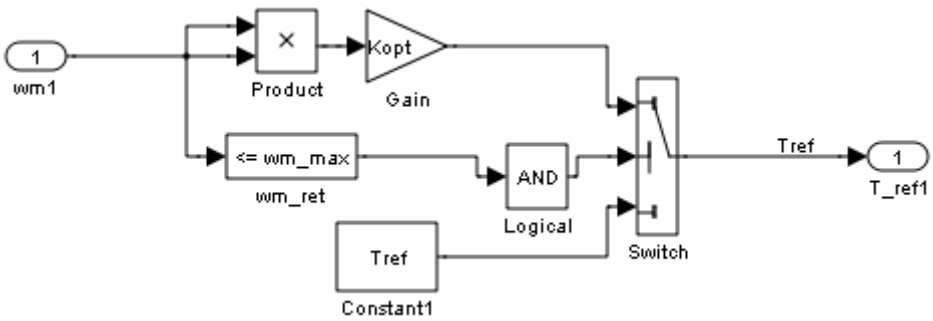


Figure B. 7: Simulink model of the reference electrical torque calculation

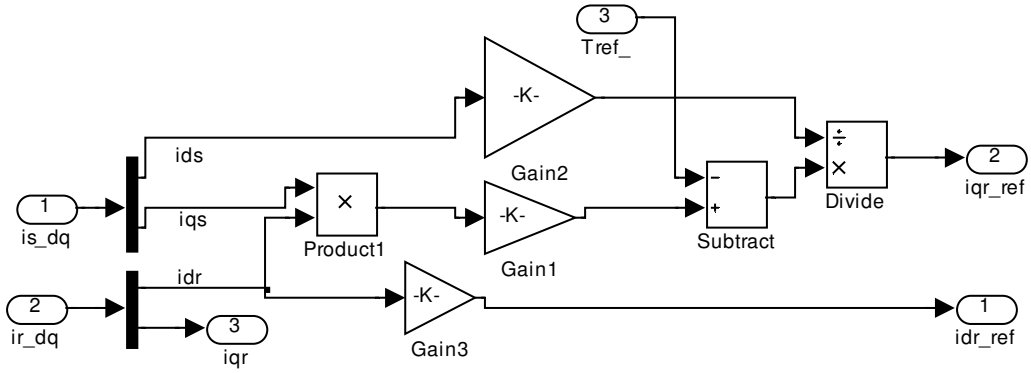


Figure B. 8: Simulink model of the reference rotor current calculation

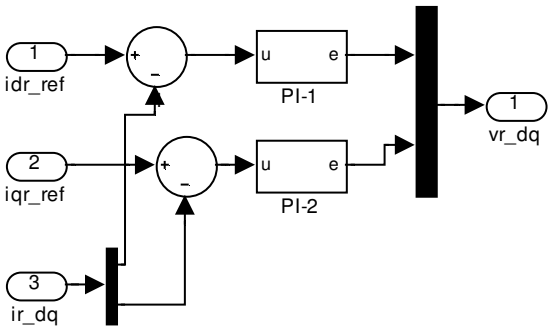


Figure B. 9: Simulink model of the rotor voltage controller

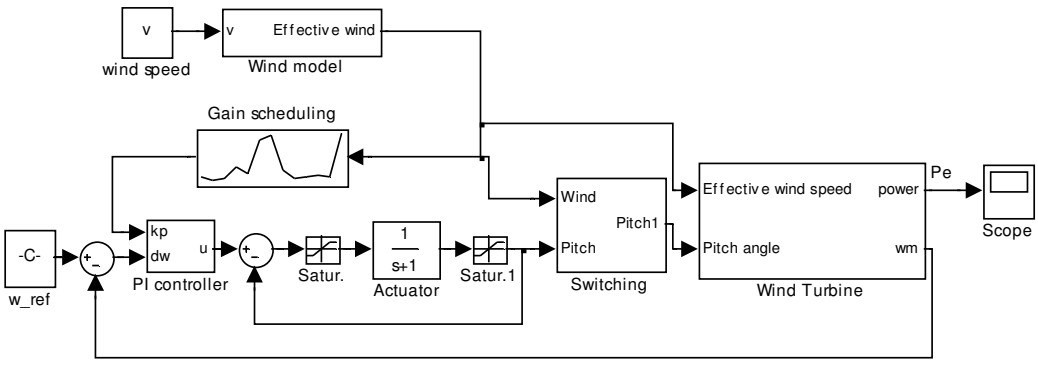


Figure B.10: Simulink model of the Pitch angle PI controller with gain scheduling

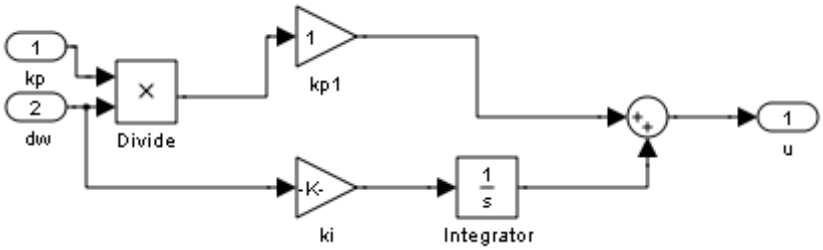


Figure B.11: Simulink implementation of PI controller

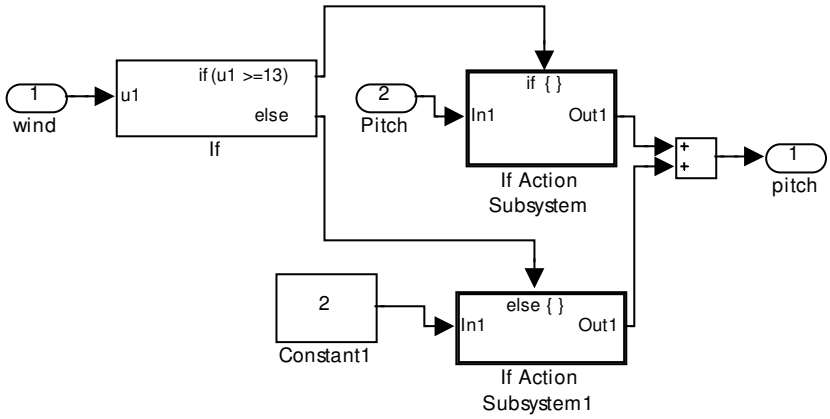


Figure B.12: Simulink implementation of switching

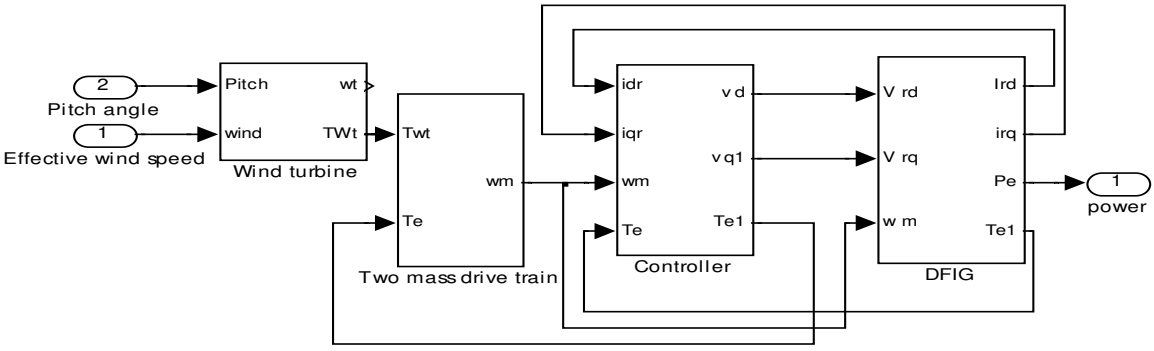


Figure B.13: Simulink model of non-linear wind turbine. In this model, we modified the models of wind turbine, DFIG and torque controller as shown in figures B.14, B.16 and B.17 respectively.

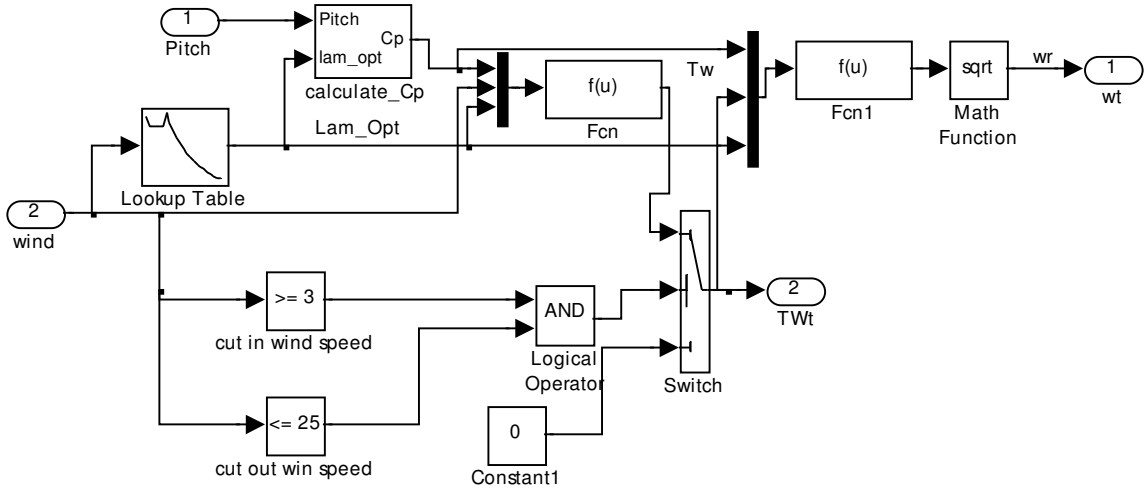


Figure B.14: Simulink block diagram model of the wind turbine. Subsystem to calculate Cp is presented in Figure B.15

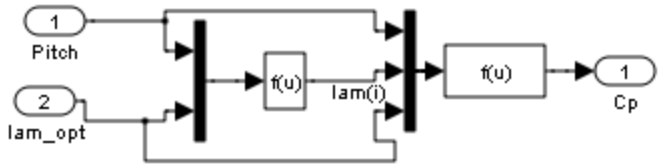


Figure B.15: Subsystem to calculate power coefficient in Figure B.14

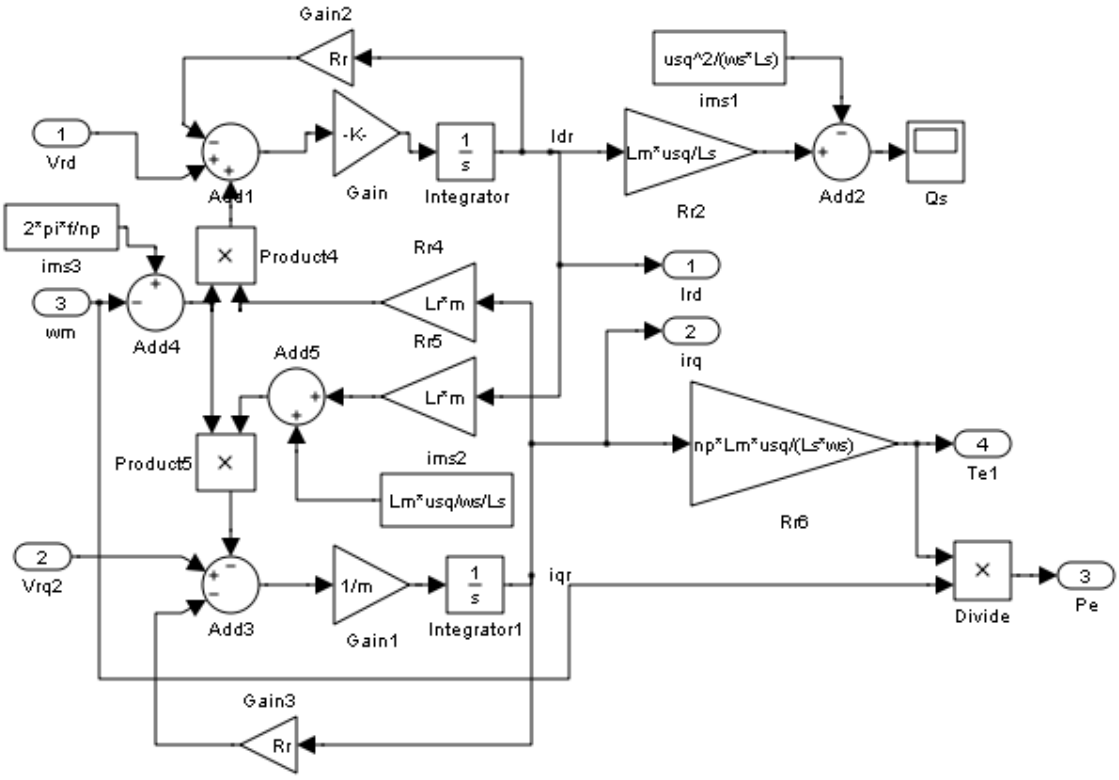


Figure B. 16: Simulink model of the DFIG in Figure B.13

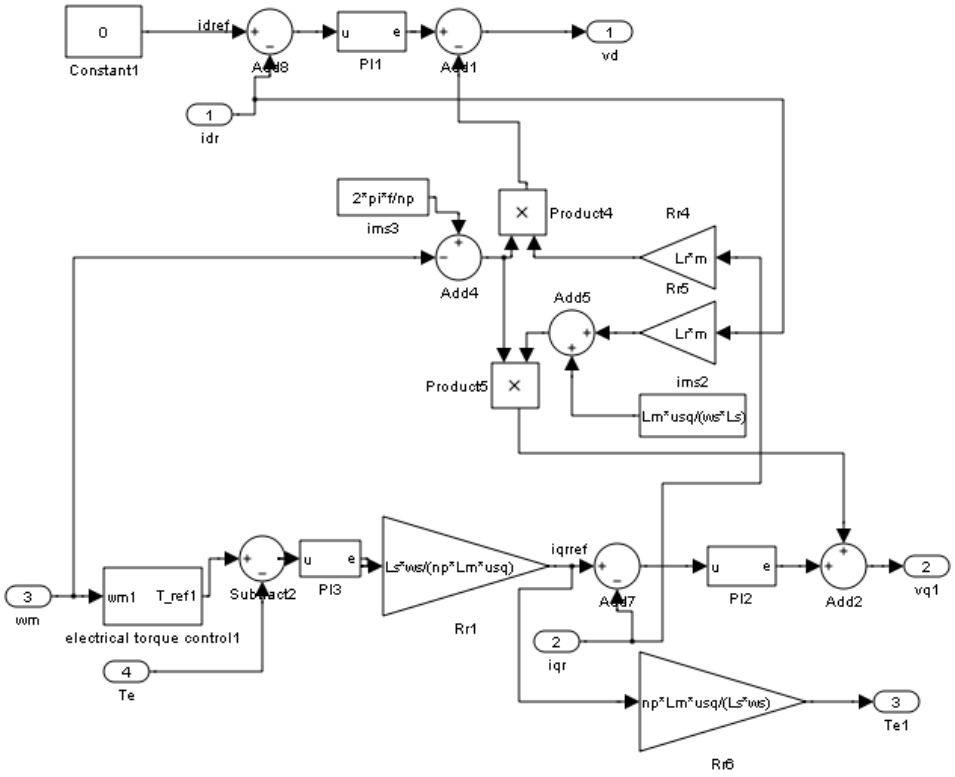


Figure B. 17: Simulink model of the electric torque controller in Figure B.13

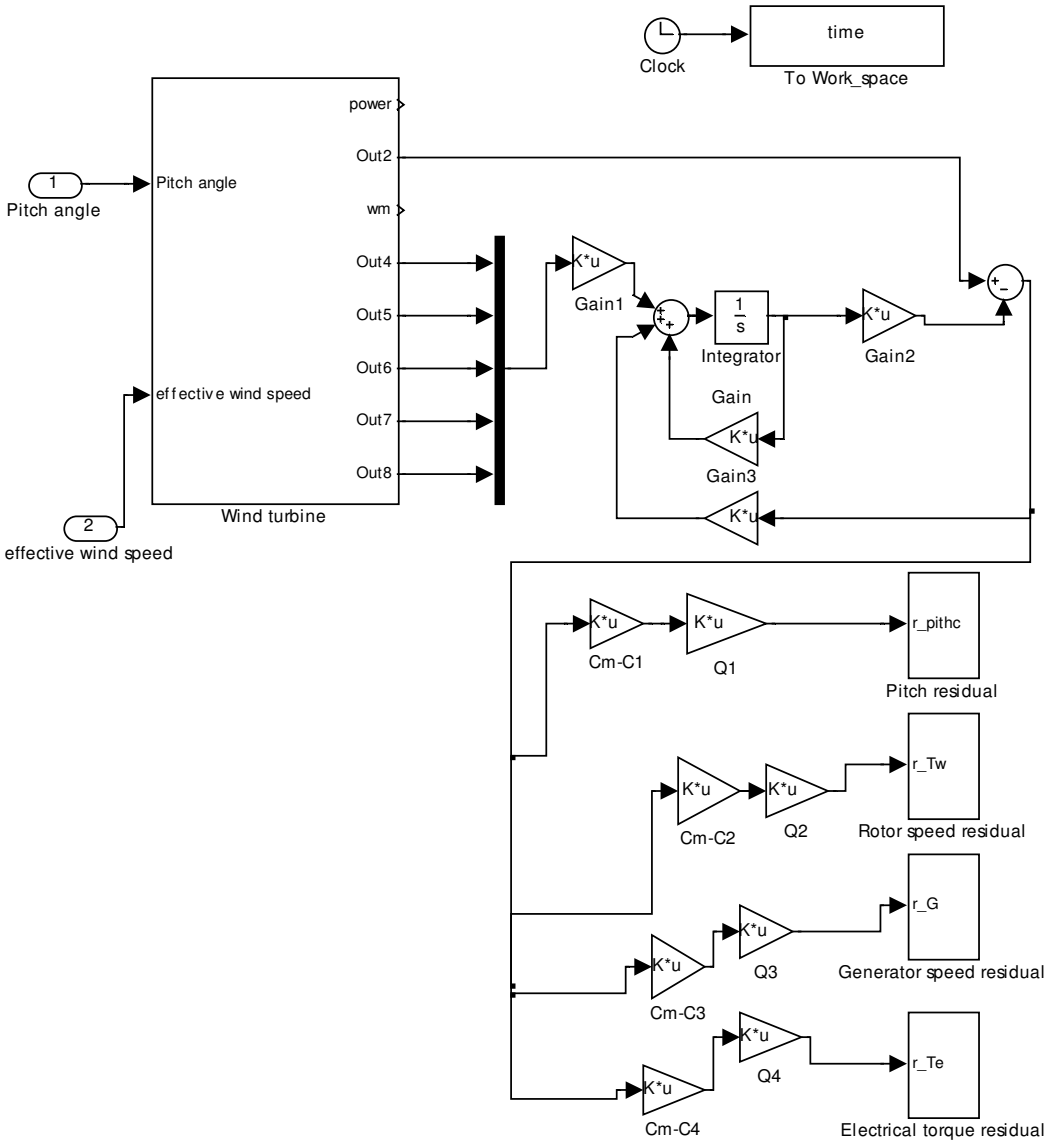


Figure B.18: Simulink model for observer-based sensor fault detection and isolation

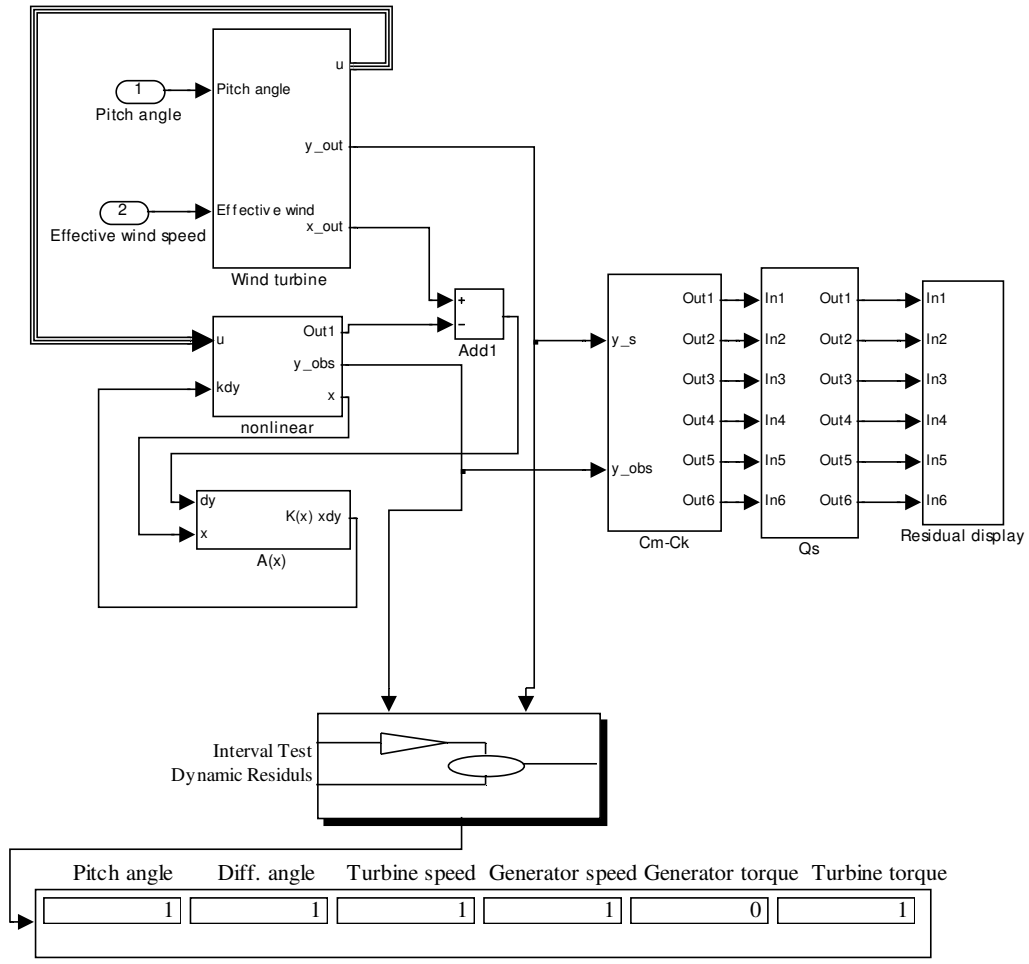


Figure B.19: Robust non-linear observer-based fault detection and isolation scheme. The simulink model of the wind turbine and interval test dynamic residual are illustrated in Figures B.22 and B.20. Here, is indicated “0” in the display of electrical torque sensor fault happened

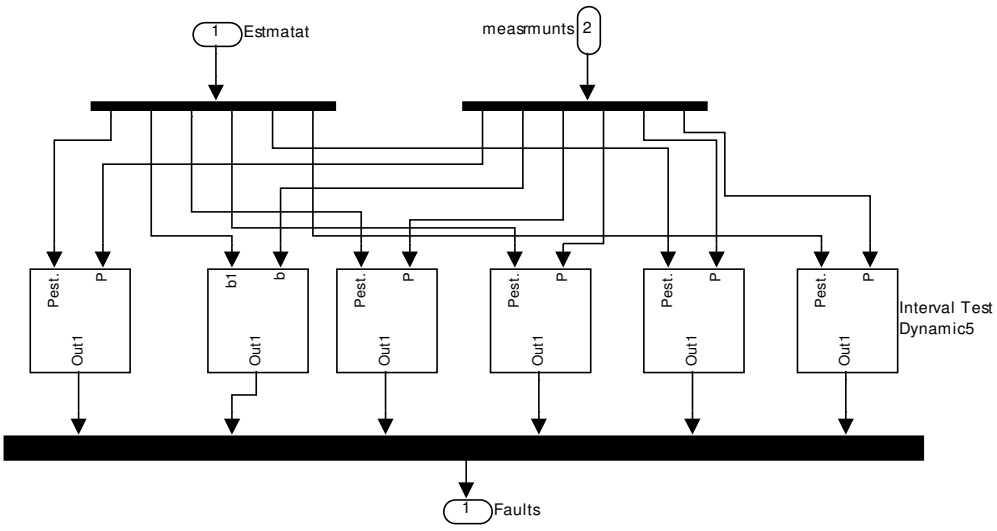


Figure B.20: The simulink model of the interval test dynamic residuals

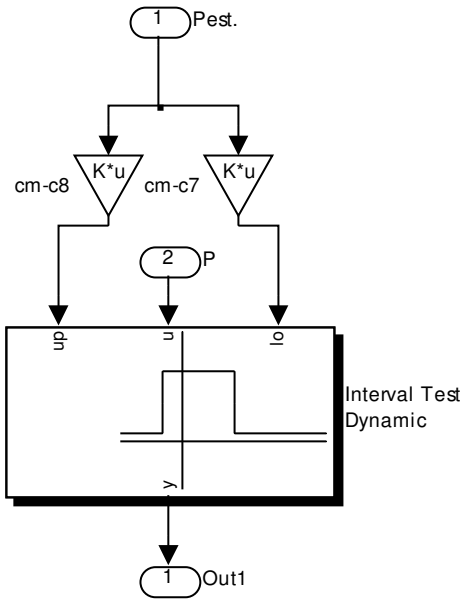


Figure B.21: The simulink model of the interval test dynamic residual of pitch angle. All other interval test dynamic residuals in Figure B.20 have the same structure

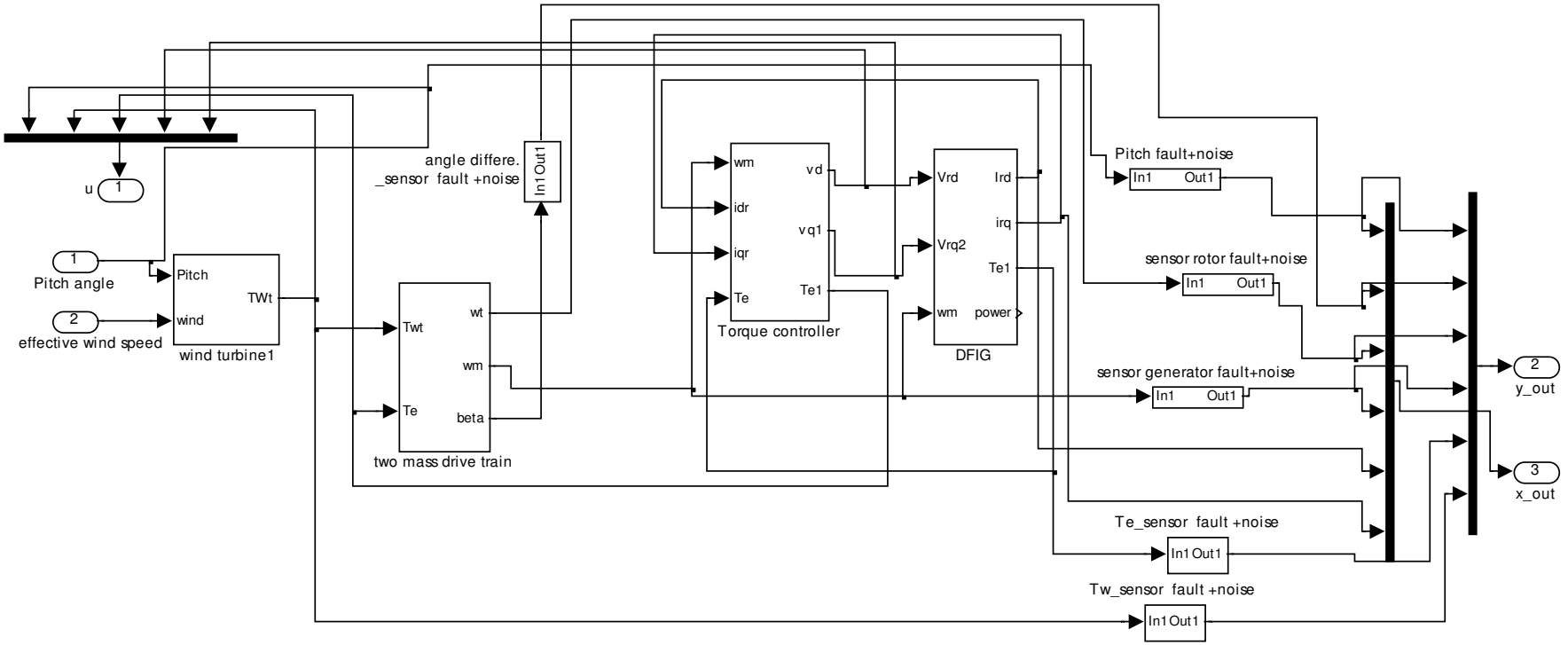


Figure B.22: Simulink model of non-linear wind turbine with faults and noises

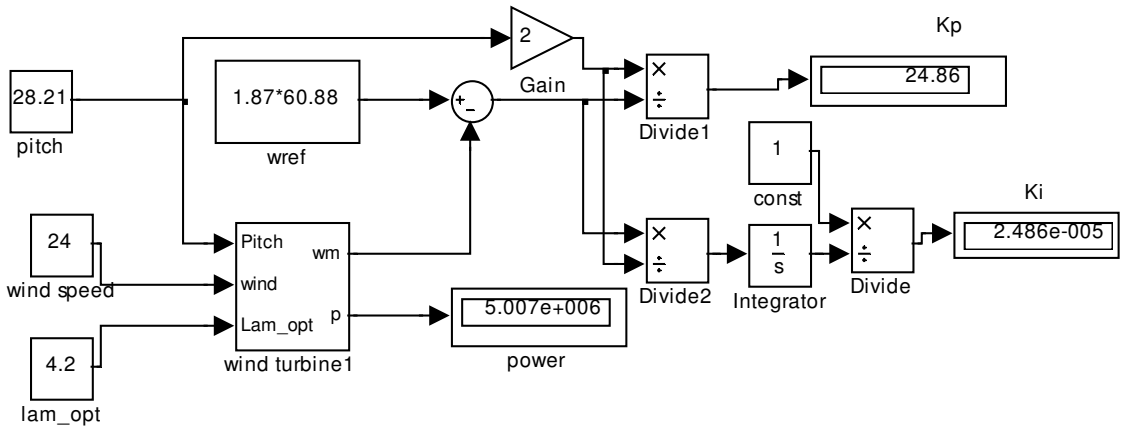


Figure B.23: Simulink model to calculate K_p and K_i

Appendix C: Multi-objective genetic algorithm

Multi-objective genetic algorithms

For multiple-objective problems, the objectives are generally conflicting, thus preventing simultaneous optimisation of each objective. Many, or even most, real engineering problems have multiple-objectives, for example to minimise cost, maximise performance, maximise reliability, etc. These are difficult but realistic problems. Genetic algorithms (GA) can be easily modified to find a set of multiple non-dominated solutions in a single run. The ability of a GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with discontinuous and multi-modal solution spaces. The crossover operator of a GA may exploit structures of good solutions with respect to different objectives to create new non-dominated solutions in unexplored parts of the Pareto front. In addition, most multi-objective GAs do not require the user to prioritise, scale or weigh objectives. Therefore, GAs have been the most popular approach to multi-objective design and optimisation problems.

I. GA Terminology

- ***Individual:***

A solution vector $x \in X$ is called an individual or a chromosome. Chromosomes are made of discrete units called genes. Each gene controls one or more features of the chromosome.

- ***Population:***

GA operates with a collection of chromosomes, called a population. The population is usually randomly initialised.

- ***Crossover:***

The crossover operator is the most important operator of a GA. In crossover, generally two chromosomes, called parents, are combined together to form new

chromosomes, called offspring. The parents are selected from existing chromosomes in the population with a preference towards fitness so that the brood is expected to inherit good genes, which make the parents fitter. By iteratively applying the crossover operator, genes of good chromosomes are expected to appear more frequently in the population, eventually leading to convergence of an overall good solution.

- ***Mutation***

The mutation operator introduces random changes into the characteristics of the chromosomes. Mutation is generally applied at the gene level. In typical GA implementations, the mutation rate (the probability of changing the properties of a gene) is very small so that new chromosomes produced by mutations will not be very different from the original chromosomes. Mutation re-introduces genetic diversity back into the population and assists the search escape from local optima.

- ***Selection***

Chromosomes for the next generation are selected. In the most general case, the fitness of an individual determines the probability of its survival for the next generation. Different selection procedures are used depending on the fitness values.

II. The procedure of a generic GA is given as follows:

1. Randomly generate a number of solutions to form the first population, P1.
   Evaluate the fitness of solutions in P1.
2. Crossover. Generate an offspring population:
   - select two solutions, *x* and *y,* from the population according to fitness values,
   - using a crossover operator, generate offspring.
3. Mutation. Mutate each solution using a predefined mutation rate.
4. Fitness assignment. Evaluate and assign a fitness value to each solution.
5. Selection. Select a number of solutions based on their fitness.
6. If the termination criterion is satisfied, terminate the search and return the current population, otherwise, set i=i+1, go to 2.

Appendix D: Reference frame conversion

I. Transformation from a three phase to a stationary reference frame

A three phase signal with three quantities (s_a, s_b and s_c), such as voltage (v), current (i) and flux ψ can be transformed to a two phase vector in the complex plane by:

$$\begin{bmatrix} s_\alpha \\ s_\beta \\ s_0 \end{bmatrix} = T_s \begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix}$$

where

$$T_s = c \begin{bmatrix} 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

c is a constant. If we take $c = \sqrt{\frac{2}{3}}$ and, then, T_S is unitary, the power of the system in the *abc*-frame is the same as in the $\alpha\beta$ - frame. Then, the $\alpha\beta$ system is called power-invariant. If we take $c = \frac{2}{3}$, then the modulus of the signal is maintained after transformation. Then, this $\alpha\beta$ system is called non power-invariant.

II. Transformation from a stationary to a rotating reference frame

$\alpha\beta$ to *dq* transformation can be written as:

$$\begin{bmatrix} s_d \\ s_q \\ s_0 \end{bmatrix} = T_R \begin{bmatrix} s_\alpha \\ s_\beta \\ s_0 \end{bmatrix}$$

where

$$T_R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the d axis makes an angle θ with a winding which has been chosen as the reference.

III. Power-invariant transformation

We assumed a sinusoidal symmetric three-phase supply voltage system with RMS value V, the d-q axis is rotating with angular velocity equal to ω_1 and phase shift 2π /3 given by:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \sqrt{2}V \begin{bmatrix} \cos(\omega_1 t) \\ \cos(\omega_1 t + \frac{2\pi}{3}) \\ \cos(\omega_1 t + \frac{4\pi}{3}) \end{bmatrix}$$

The instantaneous active power, *P*, is given by

$$P = v_\alpha i_\alpha + v_\beta i_\beta = v_d i_d + v_q i_q$$

The instantaneous reactive power, *Q*, is given by

$$Q = -v_\alpha i_\beta + v_\beta i_\alpha = -v_d i_q + v_q i_d$$

Non power-invariant transformation

The instantaneous active power, P, is given by

$$P = \frac{3}{2}(v_\alpha i_\alpha + v_\beta i_\beta) = \frac{3}{2}(v_d i_d + v_q i_q)$$

The instantaneous reactive power, Q, is given by

$$Q = \frac{3}{2}(-v_\alpha i_\beta + v_\beta i_\alpha) = \frac{3}{2}(-v_d i_q + v_q i_d)$$